2012

A Unified Nonlinear Adaptive Approach for the Fault Diagnosis of Aircraft Engines

Remus C. Avram

Wright State University

A UNIFIED NONLINEAR ADAPTIVE APPROACH FOR THE FAULT DIAGNOSIS OF AIRCRAFT ENGINES

A Thesis submitted in partial fulfillment

of the requirements for the degree of

Masters of Science in Engineering

By

REMUS CALIN AVRAM

B.S.E. University of Texas at San Antonio, 2009

2012

WRIGHT STATE UNIVERSITY

WRIGHT STATE UNIVERSITY

SCHOOL OF GRADUATE STUDIES

JANUARY 13, 2012

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Remus C Avram ENTITELED "A Unified Nonlinear Adaptive Approach for the Fault Diagnosis of Aircraft Engines" BE ACCEPTED IN PARTIAL FULLFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF Masters of Science in Engineering.

_____

Xiaodong Zhang, Ph.D

Thesis Director

_____

Kefu Xue, Ph.D
Chair, Department of Electrical Engineering

Committee on
Final Examination

_____
Xiaodong Zhang, Ph.D.

_____
Kuldip Rattan, Ph.D.

_____
Kefu Xue, Ph.D.

_____
Andrew Hsu, Ph.D.
Dean, Graduate School

ABSTRACT

Avram, Remus C. M.S.Egr., Departament of Electrical Engineering, Wright State University, 2012. A Unified Nonlinear Adaptive Approach for the Fault Diagnosis of Aircraft Engines**.**

In this thesis, a fault detection and isolation (FDI) method is developed for aircraft engines by utilizing nonlinear adaptive estimation techniques. Engine sensor faults, actuator faults and component faults are considered under one unified framework. The fault diagnosis architecture consists of a bank of nonlinear adaptive estimators. One of them is the fault detection estimator used for fault detection, and the remaining ones are fault isolation estimators employed to determine the particular fault type/location after fault detection. Each isolation estimator is designed based on the functional structure of a particular fault type under consideration. The FDI architecture has been integrated with the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) engine model developed by NASA researchers in recent years. Extensive simulation results and comparative studies are conducted to verify the effectiveness of the nonlinear FDI method.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Acknowledgements

I would like to express my sincere gratitude to my mentor Dr. Xiaodong (Frank) Zhang, for his restless attention and dedication towards my academic progress. I am also much obliged to my committee members Dr. Kuldip Rattan and Dr. Kefu Xue for both attending my defense as well as for their valuable input during my academic attendance. Moreover, my greatest thanks go to my wife, Julee L Avram for her continuous love, appreciation and support of my academic and professional aspirations.

Fault Detection and Isolation Overview

1.1 Introduction

It goes without say that the twentieth century has been the most dominant technological developed era of recorded human history. Among many great inventions and discoveries the invention of the first controlled, heavier than air aircraft, on December 17, 1903 by the Orville and Wilbur Wright, marked the beginning of an extraordinary journey in the technological evolution of human-controlled, powered flight. The driving forces of this development are the continuous growing demands in flight performance, effectiveness, safety, reliability and advanced controls and navigation systems. The strong interdependency of these constraints plays a great role in the technological advancements and expropriates high attention to design detail. With a growing demand in the safety, reliability and longevity of aircraft engine models comes the need for an accurate and continuously improving engine health-monitoring (EHM) technology. This thesis will present a model-based approach of fault detection and isolation (FDI) for generic gas-path turbine aircraft engines.

1.1.1 Model-based Fault Diagnosis and Isolation

The common definition of a fault (see book [2] for instance) is "*an unpermitted deviation of at least one characteristic property (feature) of the system from the acceptable, usual, standard condition*" (Isermann p. 20). Faults can be caused by system malfunction or external agents and can sometimes be misidentified as a result of normal

component degradation. The general guidelines for health monitoring and fault diagnosis point towards *the detection* and *the classification/isolation* of the occurring fault. A fault diagnosis system is a health monitoring system that yields as much possible information about the anomalous performance of the functioning components. Fault diagnosis (FD) carries the following responsibilities:

- *Fault Detection* - represents the process by which the presence/occurrence of a fault is identified
- *Fault Isolation* - represents the process by which the location of a fault is identified
- *Fault Identification* - represents the process by which the magnitude and nature of the fault is established

1.1.1.1 Fault Detection

As the name suggests, model-based fault diagnosis is the procedure founded on a mathematical model of the process to be diagnosed. This method is derived from the popular *hardware redundancy based fault diagnosis*, which consists of the reconstruction of the system using identical (redundant) hardware components [9]. Unlike its precursor, the model-based FD makes use of current progress in computing technology substituting redundant hardware with software implementation of the mathematical model. Hence, the alternative denominations are known as *software redundant* or *analytical redundant fault diagnosis*. The general framework of the model-based FD is shown in

Figure **1**. As seen, the process model runs in parallel with the actual process, being driven by the same inputs. Assuming enough information is known about the process, it

can be shown that the reconstructed model will mimic and approximate the real values of the process under fault-free conditions. If a fault occurs in the process, the residual (difference signal between process and process model) will be nonzero, indicating the presence of a fault. This systematization is also known as *residual generation* and is common to several model-based FD architectures [9]. Furthermore, for identification and isolation purposes *residual evaluation* is applied by means of robust thresholds designed around the expected residuals [9, 10].

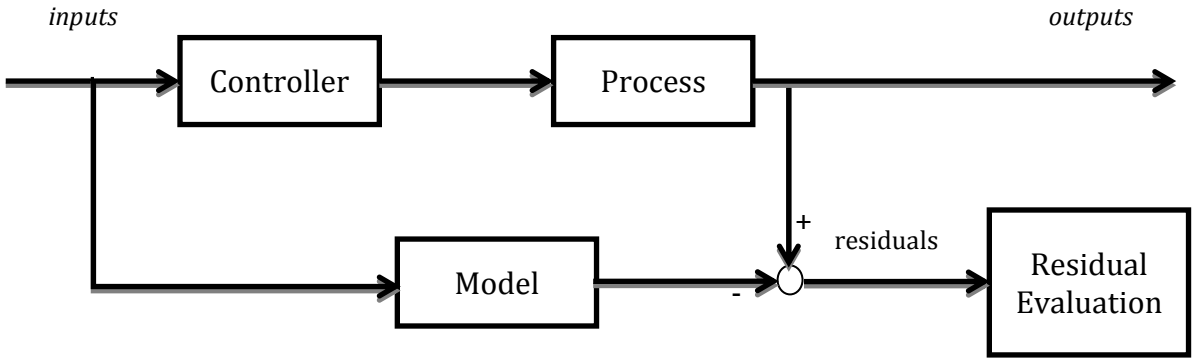


Figure 1. Model-based fault diagnosis architecture.

Model-based FD architectures have been researched and studied since the 1970's; a few paramount model-based fault diagnosis and isolation schemes stand out:

- Parameter Identification Based Methods
- Observer-based Fault Detection
- Fault Detection with Parity Equations

A. Parameter Identification Based Methods

The parameter estimation method is based on the assumptions that parameters of the plant are affected by the fault and therefore they alter their values only under abnormal operating conditions. Derivations of error minimization designs are most commonly used to satisfy this approach. Figure 2 shows such an example and considers the least squares method.

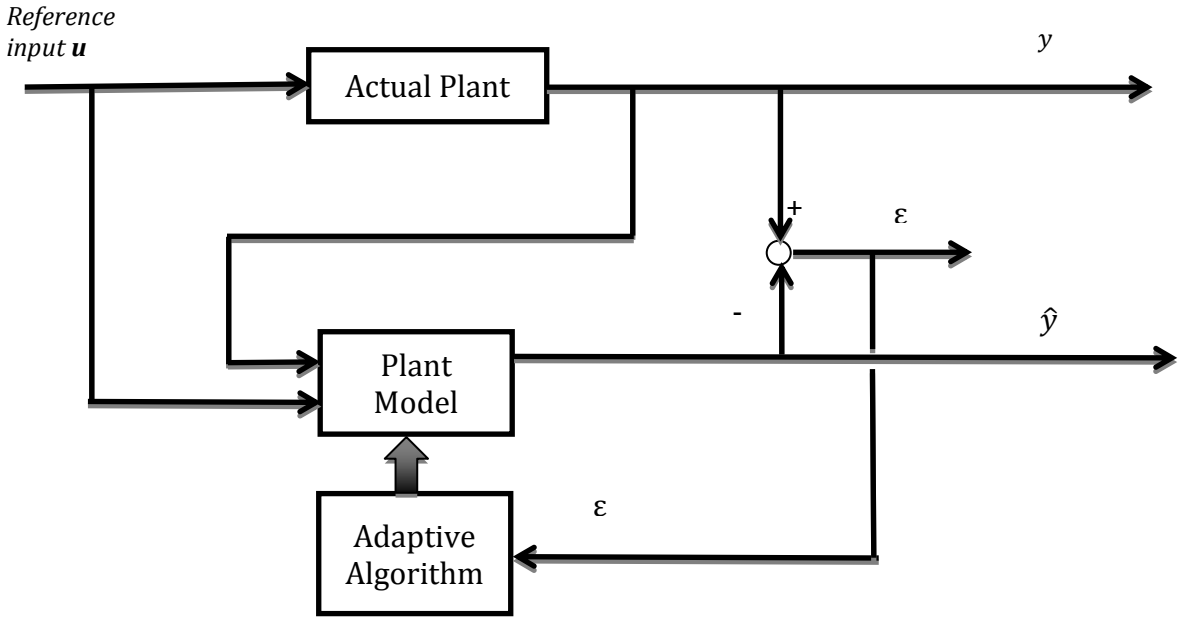


Figure 2. Parameter Estimation FD Architecture.

Let a time-invariant process be described the following difference equation:

$$y(k)+a_1y(k-1)+...+a_my(k-m)=b_1u(k-d-1)+...+b_mu(k-d-m) \quad (1)$$

Then the parameter estimation equation is given by (2):

$$\hat{\Theta}_{k+1}=\hat{\Theta}_k+\gamma(k)e(k+1), \quad (2)$$

where $\hat{\Theta}$ is the parameter estimate row vector of the parameterized measure signal $y(k) = \Psi^T(k)\Theta$, with $\Theta^T = [a_1 ... a_n \; b_0 ... b_m]$. Hence the estimate model takes the form: $\hat{y}(k) = \Psi^T(k)\hat{\Theta}$. The error signal ε is defined as: $\varepsilon(k) = y(k) - \hat{y}(k)$. In the presence of a fault, it is evident that the parameter estimation error ($\Delta\theta = \theta - \hat{\theta}$) will diverge from zero value if one of the parameters of $y(k)$ changes their value. Moreover, the nonzero elements in $\varepsilon(k) = \Psi^T(k)[\Theta - \hat{\Theta}]$ can provide some insight about the nature of the occurring fault/event. There are a series of other parameter estimation methods (recursive least square, output error equation, etc. [9,10,11]) that have been extensively researched. The most common feature shared by these methods is that they all generally require sufficient process input excitation and are especially suitable for multiplicative faults [10].

B. Fault Detection with State Observers

When a model of the process is known, state or output observer can be easily designed. Because of the specific design approach, state/output observers are commonly used for fault detection. For instance, consider the case of a sensor fault in a process represented by the state-space model given by equation (3):

$$\begin{cases} \dot{\boldsymbol{x}}(\boldsymbol{t}) = \boldsymbol{Ax}(\boldsymbol{t}) + \boldsymbol{Bu}(\boldsymbol{t}) \\ \boldsymbol{y}(\boldsymbol{t}) = \boldsymbol{Cx}(\boldsymbol{t}) + \boldsymbol{Mf_M}(\boldsymbol{t}) \end{cases} \tag{3}$$

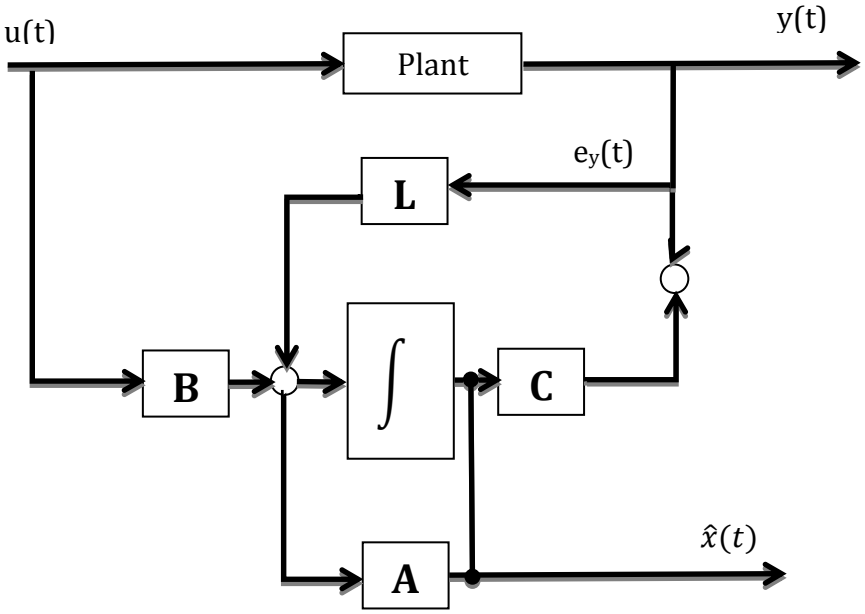


Figure 3. State Observer FD Architecture.

In equation (3) f_M represents the additive fault in the output sensor. The observer equations are given by:

$$\dot{\hat{x}}(t) = A\hat{x}(t) + Bu(t) + L[y(t) - \hat{y}(t)]$$

$$\boldsymbol{\hat{y}(t) = C\hat{x}(t)} \quad (4)$$

Hence, the output error signal is given by $e_y(t) = y(t) - \hat{y}(t)$. It is clear that when a fault occurs, the output error signal becomes nonzero. When noise and modeling uncertainties are present, state transformations are used to decouple the unknown signals.

C. Fault Detection with Parity Equations

Fault detection using parity equations, is a very similar form of the observer based FD scheme. The main difference is that FD is generally obtained based on input-output representation of the system. Consider the architecture shown in Figure 4.

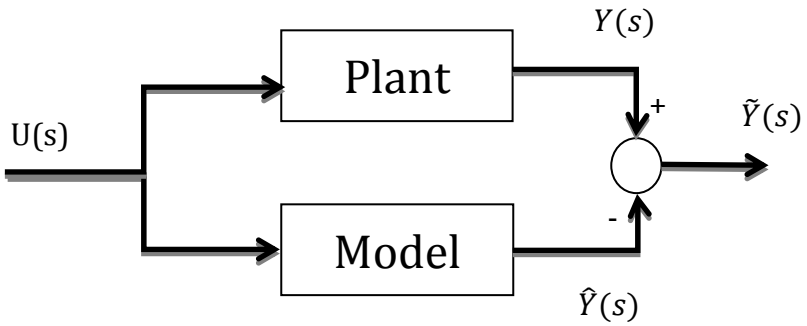


Figure 4. Parity Equation FD Architecture.

The plant dynamics are given by:

$$G_P(s) + \Delta G_P(s) = \frac{Y(s)}{U(s)} \ , \qquad (5)$$

where $\Delta G_P(s)$ represents the change in the system dynamics as a result of faults, and is equal to zero under fault free conditions. The idea behind this method, as seen in the above figure, is to run a fixed model of the plant in parallel and analyze the residuals. Under fault-free operation conditions the residuals $\tilde{Y}(s) = Y(s) - \hat{Y}(s) = [G_P(s) - G_M(s)]U(s)$ will be zero or limited by model uncertainty and noise boundary. However, when $\Delta G_P(s)$ becomes nonzero (a fault occurs) the residuals reflect the change in the system as shown in (6):

$$\tilde{Y}(s) = Y(s) - \hat{Y}(s) = [G_P(s) + \Delta G_P(s) - G_M(s)]U(s) = \Delta G_P(s)U(s) \neq 0 \qquad (6)$$

1.1.1.2 Fault Isolation and Identification

The purpose of fault isolation and identification is to provide as much information as possible about the occurring fault(s). Information extracted from residuals along with heuristic knowledge of symptoms and process enable different fault isolation and identification techniques. Fault diagnosis methods pursuing identification goals embody fault *diagnosis with classification methods* (pattern classification, Bayes classification, decision trees, neural network design, etc.) and fault *diagnosis with inference methods* (fault trees, approximate reasoning, hybrid neuro-fuzzy systems) [10, 11]. Derived from the analysis of residual signals, an array of residual-based isolation techniques can be achieved [27] :

- Structured residual set
- Fixed direction residual vector

A structured residual set is achieved by the design of a set of residuals, sensitive only to one or a sub-set of faults. The sensitivity and insensitivity between residuals and the respective faults can be simply interpreted by the means of a decision table, as shown in Table 1.

	Residual 1	Residual 2	Residual 3
Fault 1	1	0	0
Fault 2	0	1	0
Fault 3	0	0	1

Table 1. Dedicated Residual Design Truth Table Sample.

This specific approach is known as a *dedicated residual set*. As can be seen from the above table, this setup allows for the isolation of multiple faults. However, this approach

is very hard to achieve in practice, especially when limited information about the system is available. An alternative approach, to designing the residuals, known as the *generalized residual set*, is to make each residual sensitive to all but one fault. Easier to achieve in practice, this method enables only the isolation of single faults.

	Residual 1	Residual 2	Residual 3
Fault 1	1	1	0
Fault 2	1	0	1
Fault 3	0	1	1

Table 2. Generalized Residual Design Truth Table Sample.

Table 2 exemplifies on this approach. In both cases the logic can be interpreted as indicative if the respective residual lies above or below a certain threshold. For example, using the generalized residual approach, if residuals 1 and 2 lie above a set threshold, while residual 3 is bounded by the same threshold, it can be deduced that *Fault1* occurred.

Complementarily, a space representation of the residuals can also be achieved for the purpose of fault isolability. Known as the *fixed direction residual vector*, this method relies on the representation of residuals corresponding to all possible faults, as a vector in the residual space. Specifically, each residual can be represented by:

$$r_i(t) = \alpha_i(t) l_i, \tag{7}$$

where α_i is a scalar signifying the magnitude of the vector/fault, and l_i indicates the direction of the i^{th} fault in the residual subspace. Having a complete image of the fault signature space at each point in time, the isolation procedure follows to evaluate which fault vector the residual lies closest to. In the example shown in Fig. 4, the residual lies closest to the f_1 fault signature, therefore the fault f_1 is the most likely to have occurred.

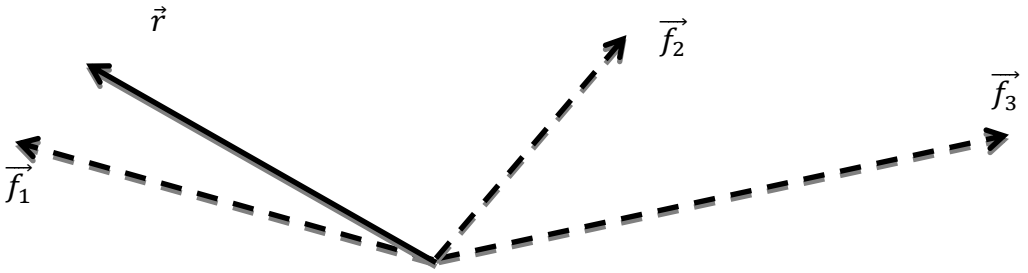


Figure 5. Fixed Direction Resdiual Vector Fault Isolation [27].

1.2 Aircraft Engine FDI Literature Review

Considerable efforts have been made in the past, starting with the 1970s, to accomplish an efficient aircraft engine health management by means of analytical redundancy fault diagnosis schemes. The constraints of fault diagnosis systems evolve around the structural analysis (fault detectability, isolability and identifiability), modeling accuracy with respect to modeling uncertainty and robustness to noise and external disturbing signals. Efforts to enhance robustness to noise, disturbances and modeling error have been widely researched [14,16, 17], most commonly advocating decoupling techniques (implemented at the residual generation stage) or the design of adaptive thresholds (implemented in the decision making scheme).

One of the earliest model-based FDI methods analyzes the system performance based on a quantifiable historical recorded flight data [6]. In Sallee *et al.* a set of recorded data is

analyzed for the purpose of studying the JT9D engine performance deterioration and its causes. The primary objectives of this method are to document historical trend of engine deterioration, quantify the levels of performance losses and identify the causes responsible. While this method can provide some insights into component interdependency and engine degradation profile, it is an offline method and thus, cannot be used as an in-flight health management monitoring system. High demand in fault detectability and isolability along with the need for online implementation pointed research towards parameter estimator techniques. Special attention has been paid to the use of Kalman Filter approaches in fault diagnostics literature. Although, various variations of observer based FD design has been accomplished [16, 19, 20, 21] and successfully implemented [17, 18], the Kalman Filter approach relies heavily on linearized models and is generally efficiently used only for sensor faults.

Efforts to capture nonlinearities in the system dynamics with enhanced algorithm stability drove the design of Kalman Filter variations, such as the *hybrid-Constant Gain Extended Kalman Filter* (*h-CGEKF*) [22, 23]. The system model is updated periodically by the means of the hybrid KF in order to adjust for engine degradation pattern. The resulting theory has been applied for a fault identification scheme of abrupt faults within sensors and actuators, using a bank of Kalman filters architecture [22, 23]. Subsequently, it has been found this approach to be susceptible to "relative large" power lever angle (PLA) movements, triggering false alarms and inconclusive in the case of PLA transients, reiterating the dependency of the KF approach on linearized model operating at steady state conditions. An alternative methodology employs the use of Neural Networks, which features fast numerical solutions once the network has been trained [24, 25]. The NN

approach reduces to a pattern recognition problem, where the patterns insinuate the different regions of the system response to different faults. Efforts to enhance multiple faults detection are shown in [26] where a bank of fault detection estimators is used. The model is comprised of N+1 FDEs, with each one corresponding to a fault signature plus a fault-free model. The analytical model is linearized based on fuzzy interpolation between operating conditions and requires extensive computational power due to its architecture.

Table 3. illustrates the benefits and drawbacks comparison of the two most common fault diagnostic techniques approaches as surveyed by Volponi *et al.* [24].

Method	**Benefits**	**Disadvantages**
Kalman Filter	• accurate and robust • historically successful • easy to configure	• requires a reasonable approximated known linear model • highly dependent on accurate input/output information
Neural Network	• exploitable when model information is scarce • once trained, provides fast and easily attainable numerical solutions	• uses influence coefficients as primary linear model • data training must be performed offline • requires retraining if engine configuration changes

Table 3. Advantages and disadvantages of KF and NN FDI architectures

1.3 Research Motivation

National Transportation Safety Board accident data covering 7,571 US-registered aircraft from 1980 to 2001, categorized by accident cause, reveals that 52 % of the hardware-induced accidents were related to aircraft system malfunctions, and 36% of these accidents were caused by propulsion system component malfunctions [28]. Advanced fault diagnosis schemes for aircraft engines can potentially help to significantly improve flight safety by enabling early detection and isolation of faulty conditions.

An important area of propulsion health management (PHM) is sensor validation. Sensor faults may lead to poor regulation or tracking performance, or even affect the stability of control system. Moreover, faulty sensor output may also cause inaccurate diagnostic/prognostic results, resulting in unnecessary replacement of system components or mission abortion. One of the most difficult problems in sensor validation is that, in many practical systems both the system components (i.e., actuators and the controlled process/plant) and sensors are prone to faults. For instance, certain propulsion system components may fail as a result of damage due to harsh operating conditions or combat. Many existing PHM methods deal with component faults and sensor faults separately. Specifically, when dealing with sensor validation, people usually assume there are no component faults; when dealing with component faults, people often assume there are no sensor faults. In the former case, a component fault may be misinterpreted as sensor faults; in the latter case, a sensor fault may be misinterpreted as a component fault. Both cases may potentially lead to a high false alarm rate and unnecessary maintenance. **Therefore, it would be ideal to address sensor faults and components faults under one unified framework.**

In addition, many existing fault detection and isolation (FDI) methods for aircraft engines are based on the assumption that the system exhibits linear behaviors in the neighborhood of steady-state operating points, and therefore linearization-based diagnostic methods are often used [30, 31, 34, 35]. However, the dynamics of aircraft engines are highly nonlinear, especially during dynamic transient operations. **Therefore, there is significant research interest in fault diagnostic methods that are directly based on the intrinsic nonlinearities of engine dynamics.**

In recent years, considerable research effort has been devoted to fault diagnosis of nonlinear systems under various kinds of assumptions and fault scenarios (see, for instance, the books [29, 27] and the references cited therein). The idea of using adaptive and learning techniques in fault diagnosis and accommodation has also been proposed. A nonlinear adaptive estimation based general methodology for robust fault diagnosis of nonlinear uncertain systems has been presented in [38, 4, 40, 41].

Based on the above discussions, underlining the strongest demands in the FDI architecture for engine health diagnostics, this thesis presents a unified (engine components and sensors) fault diagnosis and identification procedure based on the nonlinear dynamics of a jet engine model using adaptive estimation techniques.

Problem Formulation

2.1 C-MAPSS Engine Model

The NASA C-MAPSS engine model is a realistic representation of the nonlinear aerothermal dynamics of a 90,000-pound thrust class turbofan engine with high-bypass ratio and a two-spool configuration [1,32]. Figure 6 shows various components of the engine model. In this engine model, the rotating components consist of the fan, low pressure compressor (LPC) and low pressure turbine (LPT) on the fan shaft, and the high pressure compressor (HPC) and high pressure turbine (HPT) on the core shaft. The station designations are shown at the bottom of the figure. The state variables, health parameters, and actuators of the engine model are given in Table 3. Specifically, the C-MAPSS model has two state variables, including fan speed (Nf) and core speed (Nc), and provides a group of health parameters that can be adjusted to simulate normal health degradation and faulty conditions. The simulation operates with a representative closed-loop engine controller that regulates fan speed by manipulating three actuators (i.e., fuel metering valve, variable stator vane (VSV), and variable bleed valve (VBV)). The dynamics of actuators and sensors have also been considered in the simulation model [1]. Table 4 shows the complete set of sensors included in the simulation and their respective noise levels.

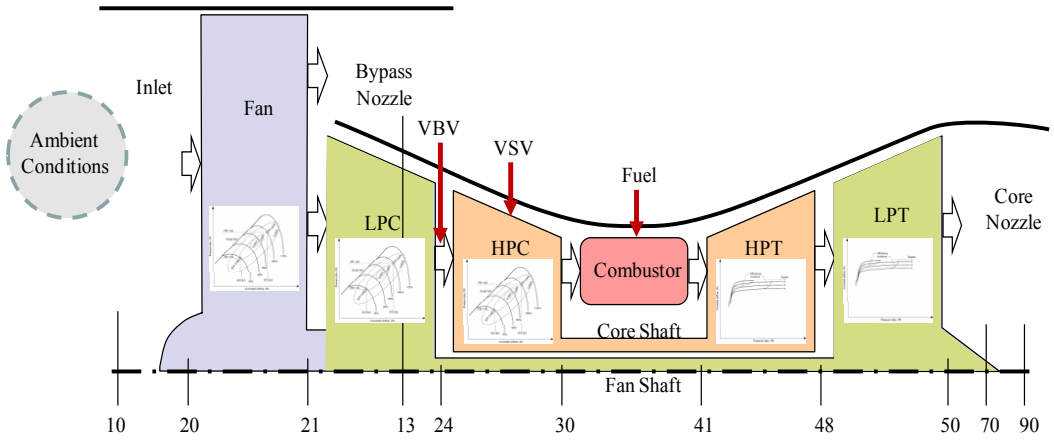


Figure 6 Diagram of the C-MAPSS Engine Components and Station Designation [4]

State Variables	Health Parameters	Actuators
Nf – fan speed Nc – core speed	Fan efficiency Fan flow capacity LPC efficiency LPC flow capacity HPC efficiency HPC flow capacity HPT efficiency HPT flow capacity LPT efficiency LPT flow capacity	Wf - fuel flow VSV - variable stator vane VBV – variable bleed valve

Table 3 State Variables, Health Parameters and Actuators

Variables	Description	Std. Deviation
Nf	Fan Speed	0.3 RPM
Nc	Core Speed	1.5 RPM
T24	LPC outlet temperature	2.2 ^{o}R
T48	HPT exhaust gas temperature	2.2 ^{o}R
Ps30	Combustor static pressure	2.8 PSI
T30	HPC outlet temperature	2.2 ^{o}R

Table 4 Sensor Descriptions and Noise

2.2 Engine System Model

The nonlinear dynamics of the C-MAPSS engine model can be described by

$$\dot{x} = Ax + \phi(x,u) + \eta(x,u,t) + \beta_x(t - T_x)f(x,u) \quad (8)$$

$$y = Cx + d(x,u,t) + \beta_x(t - T_x)\rho(x,u) + \beta_y(t - T_y)F\vartheta \quad (9)$$

where x is the state vector (i.e., Nf and Nc), u is the input vector provided by the three actuators (see Table 3), y is the output vector (see Table 4), respectively. The model given by

$$\dot{x} = Ax + \phi(x,u) \quad (10)$$

$$y = Cx \quad (11)$$

represents the known nominal engine dynamics. While the healthy system is described by

$$\dot{x} = Ax + \phi(x,u) + \eta(x,u,t) \quad (12)$$

$$y = Cx + d(x,u,t)\ . \quad (13)$$

The difference between the nominal model and the actual (healthy) system is due to the vector fields η and d, which represent the modeling uncertainties in the state equation and output equation, respectively. The changes in the system dynamics as a result of a component fault or an actuator fault are characterized by the terms $\beta_x(t - T_x)f(x,u)$ and $\beta_x(t - T_x)\rho(x,u)$ in (8) and (9) respectively. Specifically, the function $\beta_x(t - T_x)$ denotes the time profile of the component fault or actuator fault which occurs at some unknown time T_x, $f(x,u)$ and $\rho(x,u)$ represent the nonlinear fault functions in the state equation and output equation, respectively. The change in the system dynamics as a result of a sensor fault is characterized by $\beta_y(t - T_y)F\vartheta$ in (9). Specifically, the vector $F\vartheta$ represents a sensor bias, and the function $\beta_y(t - T_y)$ characterizes the sensor fault time profile with unknown fault occurrence time T_y.

The modeling uncertainty η in (8) represents the changes in the engine dynamics as a result of normal component health degradation during a number of flights. Engine normal health degradation due to usage is an aging process that all aircraft engines will experience, and it should not be considered as a fault [20]. In the fault diagnosis literature, efforts to enhance the robustness with respect to modeling uncertainty can be achieved either at the residual generation stage by using *decoupling techniques* or at the residual evaluation stage by using *adaptive thresholds* [29, 27]. In the first approach, the modeling uncertainty is often assumed to be structured, which allows the use of linear and nonlinear state transformations to decouple faults from modeling uncertainty by assuming certain rank conditions satisfied by the distribution matrix of the structured modeling uncertainty. However, in many practical systems, the modeling uncertainty is often unstructured, which makes it very difficult to apply the decoupling method. This justifies the use of adaptive thresholds to achieve robustness at the residual evaluation stage. In the adaptive threshold approach [4, 40, 41], the modeling uncertainty can be unstructured but bounded by some suitable constant or function. This bound is used to derive thresholds for distinguishing between the effect of faults and the effect of modeling uncertainty. In this research, the modeling uncertainty η, representing normal engine health degradation in a number of flights, is unstructured, and hence the second approach is adopted. Specifically, we have the following assumption:

Assumption 1. *The modeling uncertainties represented by* η *and* d *are unstructured and possibly an unknown nonlinear function of* x, u, *and* t, *but it is bounded by some known functional, i.e.,*

$$\left|\eta_i(x,u,t)\right| \le \bar{\eta}_i(y,u,t), \qquad \left|d_j(x,u,t)\right| \le \bar{d}_j(y,u,t) \tag{14}$$

where, for $i = 1, 2$ *, and* $j = 1, 2, \cdots, 6$*, the bounding function* $\bar{\eta}_i(y, u, t) > 0$ *and* $\bar{d}_j(y, u, t) > 0$ *are known, integral, and bounded for all* (x, u) *in some compact region of interest.*

As described above, the modeling uncertainty η represents the deviation of the actual engine dynamics from that of a nominal engine model, which results from normal engine component degradation during a number of flights. It is expected that such deviation of engine dynamics is small for a number of flights, since normal health degradation progresses gradually with time. Therefore, the modeling uncertainty η can be represented by small changes in certain engine component health parameters. Based on flight test data [6, 39], the bounding function on the modeling uncertainty (i.e., $\bar{\eta}_i(y, u, t)$) can possibly be obtained by utilizing the knowledge of normal degradation of the health parameters under the worst case scenario during a number of flights. It is worth noting that, normal health degradation may accumulate over time during the engine's lifetime.

As to the faults affecting the engine dynamics, the following 13 types of faults are considered in this research work:

- Component faults in fan, LPC, HPC, HPT, and LPT, respectively, which are represented by certain unknown abnormal changes in the corresponding component health parameters given in Table 3.
- Actuator faults associated with Wf and VSV, respectively, which are represented by unknown biases in the actuator output signals.
- Sensor faults in the six sensor measurements as shown in Table 4, which are represented by certain unknown biases in the sensor signals.

We assume only a single fault is possible present at any time, which could be a component fault, an actuator fault, or a sensor fault.

The class of sensor faults under consideration is represented by $F\vartheta$ in (9), where F is the fault distribution vector specifying the location of the sensor fault, and $\vartheta(t)$ is the magnitude of the possibly time-varying sensor bias. Because of the single fault assumption, the sensor fault distribution vector F has only one non-zero entry, which represents the corresponding corrupted output measurement. Depending on the location of the fault, the distribution vector F belongs to a class of six possible vectors $\{F_1, F_2, \cdots, F_6\}$, where, for any $j = 1, 2, \cdots, 6$, only the jth component of the vector F_j takes the value of 1, while all the remaining components of F_j take the value of 0.

As described above, we consider five types of component faults and two types of actuator faults, which are modeled by $f(x,u)$ in (8) and $\rho(x,u)$ of (9). Each of these seven fault types is assumed to be in the following form: for $p = 1, \cdots, 7$,

$$\begin{aligned} f^p(x,u) &= \left[\left(\theta^p(t)\right)^T g_1^p(x,u)\,, \quad \left(\theta^p(t)\right)^T g_2^p(x,u) \right]^T \\ \rho^p(x,u) &= \left[\left(\theta^p(t)\right)^T h_1(x,u)\,, \ \cdots, \ \left(\theta^p(t)\right)^T h_6(x,u) \ \right]^T \end{aligned} \tag{15}$$

where $\theta^p(t)$ represents the unknown fault magnitude, g_i^p and h_j^p are known functions representing the functional structure of the sth fault affecting the ith state equation $(i = 1, 2)$ and jth output equation $(j = 1, \cdots, 6)$, respectively. More specifically, $\theta^p(t)$ represents the unknown changes in the corresponding component health parameters (i.e., efficiency and flow capacity) in the case of a component fault and the unknown bias in the actuator output in the case of an actuator fault, respectively.

The objective of this research is to detect the occurrence of any faults and to isolate the particular fault type. Note that the cases of abrupt faults and incipient faults are both covered in the above fault model. Specifically, the fault time profile functions $\beta_x(t - T_x)$ and $\beta_y(t - T_y)$ in (8) and (9) take the form of a step function $\beta(\cdot)$ given by:

$$\beta(t - T_0) = \begin{cases} 0, & if\ t < T_0 \\ 1, & if\ t \geq T_0 \end{cases}$$

where $T_0 = T_x$ in the case of a component fault or an actuator fault, and $T_0 = T_y$ in the case of a sensor fault. Thus, the fault time profile is represented by the function $\beta(t - T_0)$, while $\vartheta(t)$ and $\theta^p(t)$ represent the (possibly time-varying) fault magnitude. For instance, in the case of foreign object damage to the fan, the function $\beta(t - T_0)$ models the sudden and immediate effect of the damage, and $\theta^p(t)$ captures the progressing effect of the fault following the initial sudden damage.

FDI Algorithm

A block diagram of the real-time engine fault diagnostic system is shown in Figure 7. Following the the general methodology in [4, 40, 41], the FDI architecture consists of a fault detection estimator (FDE) and a bank of fault isolation estimators (FIEs). Each FIE is designed based on the functional structure of a particular fault under consideration. Under normal operating conditions, the FDE monitors the system to determine the occurrence of any faults. Once a fault is detected, the FIEs are activated to determine the particular fault type that has occurred.

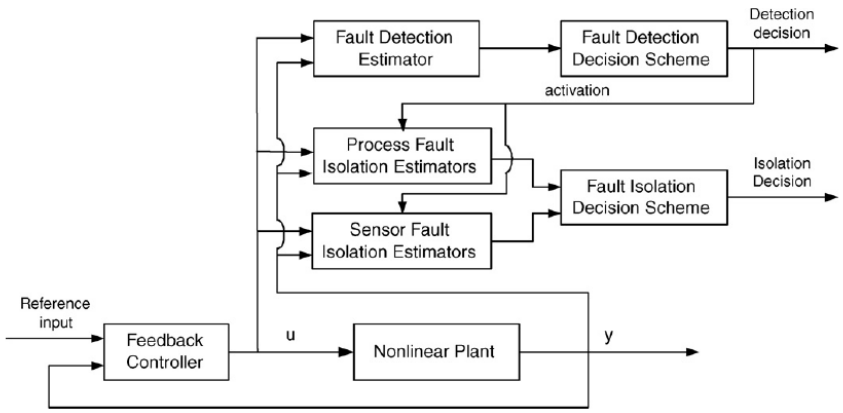


Figure 7: Block diagram of the real-time engine fault diagnostic system

3.1 Fault Detection Method

In this section, we describe the design of the fault isolation method based on the healthy system model given by (12) and (13). By filtering each side of Eq. (12) with a

stable first-order filter $\frac{1}{s+\lambda}$, where $\lambda > 0$ (In our design, we simply choose $\lambda = 1$), we obtain:

$$\frac{sx}{s+\lambda} = \frac{1}{s+\lambda}[Ax + \phi(x,u)] + \frac{\eta(x,u,t)}{s+\lambda}.$$

The above equation can be rewritten as

$$x - \frac{\lambda}{s+\lambda}x - \frac{1}{s+\lambda}[Ax + \phi(x,u)] = \frac{\eta(x,u,t)}{s+\lambda}. \tag{16}$$

By combing (13) and (16), we have

$$\bar{Y} = \begin{bmatrix} \dfrac{\eta(x,u,t)}{s+\lambda} \\ d_3(x,u,t) \\ \cdots \\ d_6(x,u,t) \end{bmatrix} \tag{17}$$

where

$$\bar{Y} = \begin{bmatrix} x - \dfrac{\lambda}{s+\lambda}x - \dfrac{1}{s+\lambda}[Ax + \phi(x,u)] \\ y_3 - C_3 x \\ \cdots \\ y_6 - C_6 x \end{bmatrix} \tag{18}$$

and C_3 and C_6 represent the 3rd row and 6th row of the output matrix C, respectively.

In the absence of faults, the state variable x (i.e., Nf and Nc) in (18) can be implemented using the corresponding sensor measurement. In addition, the term $Ax + \phi(x,u)$ can be

obtained from a nominal engine model with no degradations (see (10 and (11)) even though its analytical form is not available. Therefore, the fault detection residual $\bar{Y}$ defined by (18) can be easily implementable. Based on (14) and (17), for each residual component, we have $|\bar{Y}_k(t)| \leq \nu_k$ with $k = 1, \cdots, 6$, where

$$\nu_k = \begin{cases} \dfrac{\bar{\eta}_k}{s+\lambda}, & \text{for k} = 1, 2 \\ \dfrac{\bar{\mathrm{d}}_{\mathrm{k}}}{s+\lambda}, & \text{for k} = 3, \cdots, 6. \end{cases} \tag{19}$$

According to the above discussions, we have the following

Fault Detection Decision Scheme: A fault is detected when at least one component of the modulus of the fault detection residual $(i.e., |\bar{Y}_k(t)|)$ exceeds its corresponding threshold ν_k. The fault detection time T_d is defined as the first time instant such that $|\bar{Y}_k(t)| > \nu_k$, for some $t \geq 0$ and some $k \in \{1, \cdots, 6\}$.

3.2 Fault Isolation Method

Now assume a fault is detected at $T_d \geq T_0$, the bank of FIEs is activated for the purpose of fault isolation. Each FIE is designed based on the functional structure of a particular fault under consideration. Below we describe the design of FIEs for component faults, actuator faults, and sensor faults, respectively.

3.2.1 Component and Actuator Faults Isolation Estimators

Based on the system model given by (8) and (9), in the presence of a component fault or an actuator fault as defined by (15), the system dynamics become

$$\dot{x} = Ax + \phi(x,u) + \eta(x,u,t) + \beta_x(t - T_x)f(x,u)$$

$$y = Cx + d(x,u,t) + \beta_x(t - T_x)\rho(x,u) \ .$$

Based on (15) and by using similar techniques as reported in the fault detection method, in the presence of pth fault type, we have

$$\bar{Y} = G^p(x,u)\theta^p + \begin{bmatrix} \frac{1}{s+\lambda}\eta \\ d \end{bmatrix} \tag{20}$$

where $\bar{Y}$ is defined in (18), and $G^p(x,u) = \frac{1}{s+\lambda}\begin{bmatrix} g^p(x,u) \\ h^p(x,u) \end{bmatrix}$ with $g^p(x,u) = \begin{bmatrix} \left(g_1^p(x,u)\right)^T \\ \vdots \\ \left(g_n^p(x,u)\right)^T \end{bmatrix}$ and $h^p(x,u) = \begin{bmatrix} \left(h_1^p(x,u)\right)^T \\ \vdots \\ \left(h_l^p(x,u)\right)^T \end{bmatrix}$. Therefore, based on adaptive estimation techniques [31, 33], the following FIEs corresponding to component faults and actuator faults are chosen: for $p = 1, \cdots, 7$,

$$\hat{Y}^p(t) = G^p(x,u)\hat{\theta}^p(t) \tag{21}$$

where $\hat{\theta}^p(t)$ is the estimated fault magnitude ($\hat{\theta}^p \in \Re^2$ for component faults, and $\hat{\theta}^p \in \Re$ for actuator faults) associated with the pth FIE .

The adaptation in the FIEs arises due to the unknown fault magnitude θ^p. The following adaptive law for adjusting $\hat{\theta}^p$ is derived by using the Lyapunov synthesis method (see, for instance, [31, 33]) :

$$\dot{\hat{\theta}}^p = \Gamma G^p(\bar{Y} - \hat{Y}^p) \tag{22}$$

where Γ is a symmetric and positive-definite learning rate matrix. In the presence fault type p, the learning algorithm given by (22) ensures the residual $\tilde{Y}^p = \bar{Y} - \hat{Y}^p$ generated by the matched pth FIE converges to zero in the absence of modeling uncertainty. In the presence of modeling uncertainty, robust adaptive learning algorithms such as projection modification and σ-modification may be applied to the adaptive law (22) to ensure robustness [31, 33].

3.2.2 Sensor Faults Isolation Estimators

Based on the system model given by (8) and (9), in the presence of a sensor fault, the system dynamics become:

$$\begin{aligned} \dot{x} &= Ax + \phi(x,u) + \eta(x,u,t) \\ y &= Cx + d(x,u,t) + F\vartheta. \end{aligned} \tag{23}$$

Now, let us first consider the case of a sensor fault in Nf or Nc. Based on the generalized observer scheme [29, 27], by utilizing all sensor signals except the Nf signal, the FIE corresponding to Nf sensor fault is designed as :

$$\dot{\hat{x}} = A\hat{x} + \phi(\hat{x}, u) + L(y^1 - \hat{y}^1)$$
$$\hat{y}^1 = C_1 x \tag{24}$$

where $\hat{x}$ is the estimated state vector, $y^1 \in \Re^5$ consists of all the sensor signals except the Nf signal (i.e., Nc, T24, T48, Ps30, and T30), and $\hat{y}^1$ is the estimate of y^1, C_1 consists of all rows of C except the first row which corresponds to the Nf sensor signal, the observer gain L is chosen such that (24) is an asymptotically stable observer for the nonlinear system (23) in the absence of faults and modeling uncertainty [36]. Since the Nf sensor signal is not used to drive the above observer, the output estimation error $\tilde{y}^1(t) = y^1(t) - \hat{y}^1(t)$ should not be affected by the Nf sensor fault. In other words, in the presence of a fault in the Nf sensor, the residual $\tilde{y}^1(t)$ should still converge to zero in the absence of modeling uncertainty (or remain low in the presence of modeling uncertainty). On the other hand, in the presence of any other type faults under consideration, the residual $\tilde{y}^1(t)$ should be high since such faults will affect the observer dynamics or driving signals.

Analogously, the FIE corresponding to the Nc sensor fault can be designed as a nonlinear observer utilizing all the sensor signal except the Nc signal. The corresponding residual $\tilde{y}^2(t) = y^2(t) - \hat{y}^2(t)$ should be sensitive to all fault types except the Nc sensor fault.

Additionally, the FIEs corresponding to the remaining four types of sensor faults (i.e., T24, T48, Ps30, and T30) can be designed using a simpler method, since these signals do

not belong to the state vector. Specifically, based on (23) and by using similar techniques as reported in the fault detection scheme, it can be easily shown that

$$\bar{Y} = \begin{bmatrix} \frac{1}{s+\lambda}\eta \\ d \end{bmatrix} + \begin{bmatrix} 0 \\ F\vartheta \end{bmatrix}$$

where $\bar{Y}$ is defined in (18), and the vector F takes the value of $[0,0,1,0,0,0]^T$, $[0,0,0,1,0,0]^T$, $[0,0,0,0,1,0]^T$, or $[0,0,0,0,0,1]^T$. Obviously, in the presence of any of these four sensor faults, only one component of $\bar{Y}$ would be affected. Thus, for the residual $\tilde{y}^q$ $(q = 3,\cdots,6)$ corresponding to these four sensor faults, we simply choose $\tilde{y}^q(t) = \bar{Y}^q$, where $\bar{Y}^q$ consists of all components of the vector $\bar{Y}$ described by (18) except the qth component.

To summarize, based on the above design, for $q = 1,\cdots,6$, the residual $\tilde{y}^q$ is only insensitive to the qth sensor fault.

3.2.3 Fault Isolation Decision Scheme

Now let us consider the pth component or actuator fault, where $p = 1,\cdots,7$, and the qth sensor fault, where $q = 1,\cdots,6$, in a unified framework. Then, we have 13 types of faults in the fault set. More specifically, for $s = 1,\cdots,13$, fault s is a component fault or an actuator fault, if $1 \leq s \leq 7$, and fault s is a sensor fault, if $8 \leq s \leq 13$. Thus, the fault isolation residuals $\tilde{Y}^p(t)$ (for component and actuator faults) and $\tilde{y}^q(t)$ (for sensor faults)

can be represented under the unified framework as $\epsilon^s(t)$, $s = 1, \cdots, 13$, where $\epsilon^s(t) = \tilde{Y}^s(t)$, if $s = 1, \cdots, 7$; and $\epsilon^s(t) = \tilde{y}^{s-7}(t)$, if $s = 8, \cdots, 13$.

Due to the presence of modeling uncertainty, residuals are never zero even in the absence of faults. Moreover, the residual may change with the time-variant control inputs and dynamic engine operating conditions. Consequently, a small fixed threshold may result in false alarms, while a large fixed threshold may increase the number of missed detections/isolations. In the research, adaptive thresholds developed in [4, 40, 41] are utilized to enhance fault sensitivity and robustness. The design of adaptive thresholds for fault isolation is briefly described below. The detailed analysis can be found in [4, 40, 41].

As described in [4, 40, 41], the adaptive thresholds for each FIE are designed in such a way that in the presence of a particular fault type s, where $s = 1, \cdots, 13$, the residuals generated by all FIEs may exceed their corresponding thresholds, except for the matched sth FIE. More specifically, the fault isolation scheme is based on the following principle: for each particular fault s, a set of adaptive threshold functions $\mu_j^s(t)$ can be designed, such that the jth component of the residual vector $\epsilon^s(t)$ generated by the sth FIE satisfies $\epsilon_j^s(t) \leq \mu_j^s(t)$, for all $j = 1, \cdots, 6$. Therefore, in the fault isolation procedure, if for a particular FIE r, at least one component of the corresponding residual vector exceeds its threshold (i.e., $\epsilon_j^r(t) > \mu_j^r(t)$ for some $j = 1, \cdots, 6$), then the case of fault type r having occurred can be excluded.

Based on this intuitive idea, we have the following

Fault isolation decision scheme: The occurrence of fault type s is concluded if the following two conditions are both satisfied:

a) All the diagnostic residuals, generated by FIE s, remain below their corresponding adaptive thresholds (i.e., $\epsilon_j^s(t) \leq \mu_j^s(t)$, for all $j = 1, \cdots, 6$).

b) At least one component of the diagnostic residuals generated by any remaining FIE r, where $r \in \{1, \cdots, 13\} \backslash \{s\}$, exceeds its corresponding threshold, i.e., $\epsilon_j^r(t) > \mu_j^r(t)$, for some $j \in \{1, \cdots, 6\}$.

The proposed fault isolation decision logic follows the *generalized observer scheme* (GOS) (Chen and Patton, 1999). Note that the residual generated by the matched isolation estimator is often the smallest one. Thus, the following isolation scheme is actually implemented: The fault type corresponding to the FIE with the smallest residual is considered to be the one that has occurred. Additionally, for any FIE, if at least one component of its residual exceeds the corresponding adaptive threshold, then its residual is set to a high value, hence excluding the occurrence the corresponding fault type.

Implementation

4.1 Algorithm Implementation

A prototype FDI software was implemented and integrated into the C-MAPSS engine model. As shown in Figure 8, the prototype software consists of one fault detection estimator (FDE) and 13 FIEs, one for each particular fault mode.

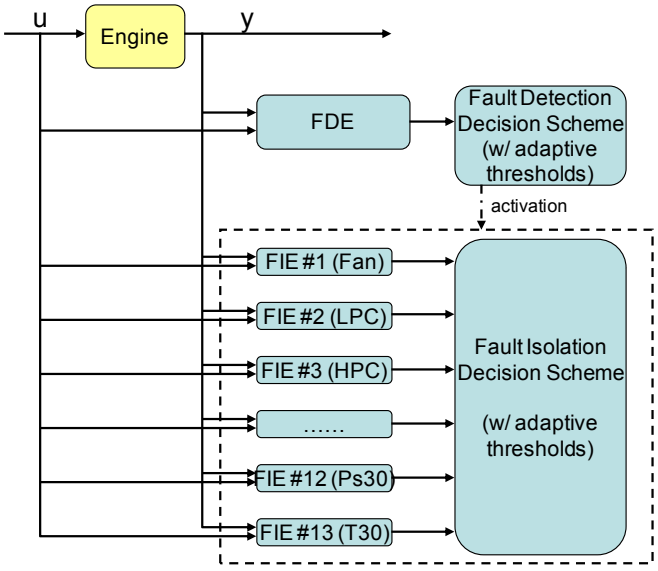


Figure 8: Components of the prototype FDI software tool

Below we briefly discussed several additional technical issues involved in the implementation of the real-time fault diagnostic algorithm. First, it is not possible to obtain the accurate mathematical equations describing the normal engine dynamics, since such analytical forms does not exist in numerical engine simulations such as C-MAPSS. Specifically, the internal dynamics of the engine *Simulink* model is very complicated, and its analytical forms are not available for fault diagnostic design. What can be changed are the engine model's input signals and health parameters. Thus, the implementation is carried out by using additional engine *Simulink* models with modified inputs and health

parameters. For instance, the nominal dynamics of a new engine is obtained numerically by using an additional engine model running in parallel with non-degraded health parameters (i.e., with all the health parameter degradations set to zero) in open-loop mode.

The modeling uncertainty resulting from normal engine degradation also affects the diagnostic residuals. A bounding function $\bar{\eta}$ on the modeling uncertainty η in (1) is required to implement the adaptive threshold for fault detection and isolation. The bounding function $\bar{\eta}$ can be obtained by utilizing the knowledge of possible normal degradation of health parameters under the worst case scenario during a number of flights. Based on historical test data [6,39], a 0.2% shift in each health parameter is considered to obtain the bounding function $\bar{\eta}$. Since it is not possible to obtain the exact analytical form of $\bar{\eta}$, it is obtained numerically based on two engine models running in parallel: one engine model with non-degraded health parameters (i.e., no modeling uncertainty) and the other engine model with 0.2% degradation (representing worst-case degradation in a number of flights) in selected health parameters (note that some health parameters may have canceling effects). As shown in Figure 9, x_N and x_D represent the state vectors generated by the engine model with non-degraded health parameters and the engine model with worst case degradation, respectively.

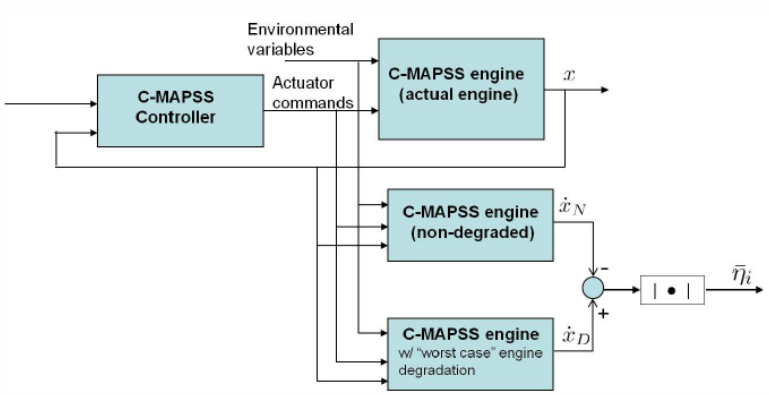


Figure 9 Method for generating the modeling uncertainty bound $\bar{\eta}$

The C-MAPSS native environment is Matlab© (*The Mathworks Inc.*). Figure 10 shows the implementation of the FDI block with the original C-MAPSS construct, as it appears in *Simulink*. As it can be seen the system is comprised of three main blocks: engine controller, engine model and the FDI block.

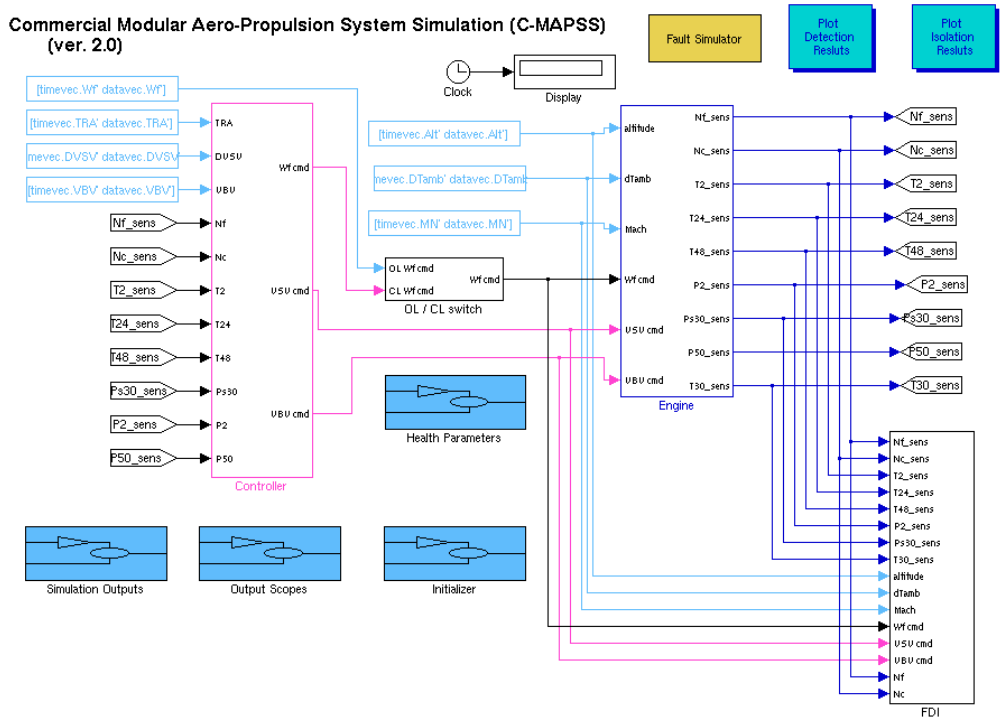


Figure 10. Implementation of the FDI Block with the C-MAPSS System

Simulation Results

Extensive simulation studies have been conducted to validate the performance of the FDI algorithm. In particular, we have also implemented a least-squares-based fault classification algorithm [37] with help from NASA researchers and conducted various comparative studies between the nonlinear adaptive FDI method and the least-squares-based fault classification algorithm.

5.1 Least-Squares-Based Fault Classification Algorithm

The least-squares-based fault classification algorithm assumes a linear relationship in the form of

$$\tilde{y} = Hx + v \ ,$$

where $\tilde{y}$ is the residual representing the deviation between the sensor measurement and its estimate obtained based on a linear engine model (see Figure 11) , x is the unknown fault magnitude, H is a fault influence matrix, and v represents normally distributed measurement noise with zero mean. Therefore, the least squares algorithm can be used to estimate the unknown fault magnitude given the residual $\tilde{y}$ and the fault influence matrix H [37]. The fault hypothesis corresponding to the smallest output estimation error is classified as the fault type that has occurred. The fault influence coefficient matrix H obtained using the fault detection residuals is shown in

Table 5.

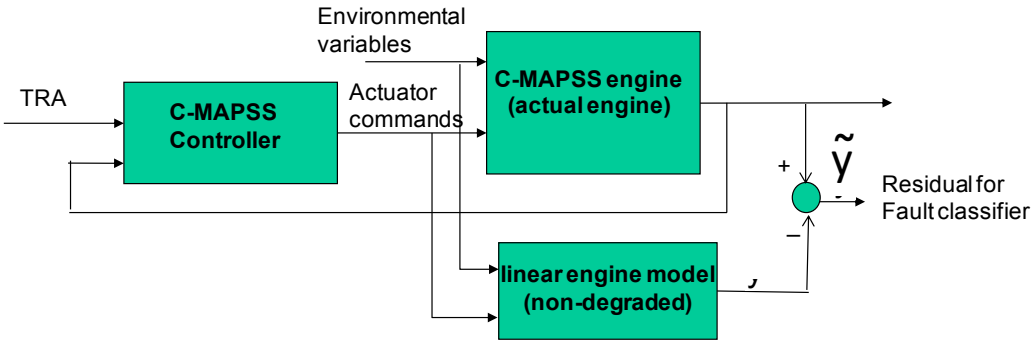


Figure 11 Block diagram used to generate residual for the fault classification algorithm

Specifically, each column of Table5(a) corresponds to the steady-state values of the residual $\tilde{y}$ in the presence of certain deviation in a particular health parameter under consideration (listed in the first row), and each column of Table 5(b) corresponds to the steady-state value of the residual in the presence of a bias in a particular actuator or sensor under consideration (listed in the first row). The simulations are conducted at the operating condition of 36K ft, Mach 0.7, dTamb = 0, and PLA = 60 degree. Specifically, in order to generate the H matrix, an efficiency loss of 1% or a flow capacity loss of 1% are assumed for Fan, LPC and HPC, respectively; an efficiency loss of 1% or a flow capacity change of 1% are assumed for HPT and LPT, respectively; a bias of -1 and -36.49 (approximately 1% of the steady state value) are used to generate the H matrix columns corresponding to actuator faults in VSV and Wf, respectively; a bias of approximately 1% of the steady-state value is used to generate the corresponding H matrix column for each sensor fault. The residuals are normalized with respect to their steady-state values. Moreover, each entry is multiplied by 100 to represent percentage of change.

Table 5**:** Fault influence coefficient matrix

(a) Fault influence coefficient matrix for components faults

	Fan Eff	Fan Flow	LPC Eff	LPC Flow	HPC Eff	HPC Flov	HPT Eff	HPT Flov	LPT Eff	LPT Flow
Nf	-0.3142	0.5606	-0.0274	0.0132	-0.2307	-0.0319	-0.3271	-0.0640	-0.3350	0.0036
Nc	0.1124	-0.0517	0.0250	0.0471	-0.5841	0.5178	-0.8429	-0.1900	0.0918	0.4383
T24	-0.0710	0.1302	0.0532	-0.0975	-0.0066	0.0034	-0.0073	-0.0010	-0.1567	-0.0664
T48	0.0371	0.0920	0.0460	0.0089	0.6151	0.0678	0.9126	0.1890	-0.0267	-0.4322
Ps30	-0.0719	-0.0562	-0.0279	-0.0519	-0.7353	-0.0551	-1.1996	-1.4050	-0.0381	0.5371
T30	0.0303	0.0453	0.0483	-0.0271	0.2774	0.0870	-0.4448	-0.3212	-0.0350	0.1917

(b) Fault influence coefficient matrix for actuator and sensor faults

	VSV	Wf	Nf	Nc	T24	T48	Ps30	T30
Nf	-0.0252	-0.3143	0.9639	-0.0178	-0.0432	0.0000	0.0000	0.0000
Nc	1.5346	-0.2302	0.0531	0.7139	0.0616	0.0000	0.0000	0.0000
T24	0.0107	-0.0987	-0.0047	0.0053	0.9891	0.0000	0.0000	0.0000
T48	0.0218	-0.3320	0.1041	0.0310	0.1223	0.9978	0.0000	0.0000
Ps30	0.0420	-0.6877	-0.1329	-0.0733	-0.1343	0.0000	0.9965	0.0000
T30	0.1562	-0.2223	0.0289	0.0041	0.0461	0.0000	0.0000	1.0000

5.2 Comparative Studies

We have conducted extensive comparative studies at various fault scenarios to verify the performance of the nonlinear adaptive FDI method. Specifically, in the comparative studies, the TRA is fixed at 60 degree. The fault occurrence time is assumed to be at 25 second. The range of fault magnitudes considered is shown in Table 6. For the purpose of comparative studies, we define the following performance metrics:

Percentage of Correct Diagnosis = (number of correct diagnosis) / (total number of diagnosis).

Note that the online real-time diagnosis routine conducts fault diagnosis at every time step. Thus, the percentage of correct diagnosis should be 100% for a perfect diagnostic algorithm.

Table 6 Ranges of fault magnitude considered

	Fan	LPC	HPC	HPT	LPT
Eff loss (%)	[1, 7]	[1, 7]	[1, 7]	[1, 7]	[1, 7]
Flow capacity loss (%)	[1, 7]	[1, 7]	[1, 7]	[-7, -1]	[-7, -1]

	VSV	Wf	Nf	Nc
Bias	$\pm[1,3]$	$\pm[40,200]$	$\pm$ [9, 90]	$\pm$ [10, 100]

	T24	T48	Ps30	T30
Bias	$\pm[2,20]$	$\pm$ [7, 56]	$\pm$ [1.5, 9]	$\pm[3,30]$

Table 7 Labels of different fault types

Fault	Fan	LPC	HPC	HPT	LPT	VSV	Wf	Nf	Nc
Label	1	2	3	4	5	6	7	8	9

Fault	T24	T48	Ps30	T30
Label	10	11	12	13

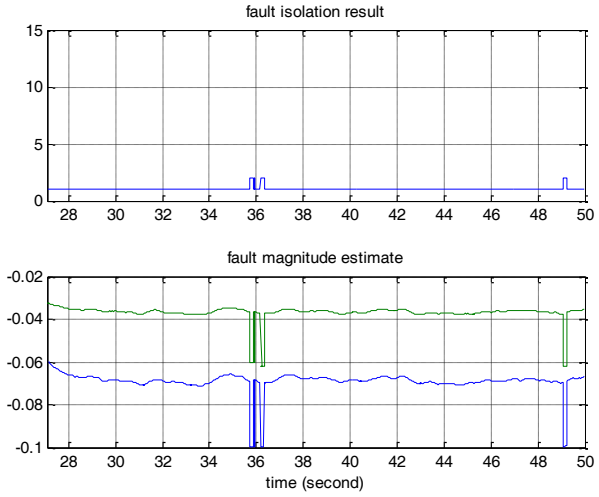
Two cases are considered below: the case without modeling uncertainty resulting from normal degradation and the case with 0.2% deviation in each health parameter.

5.2.1 Case without Modeling Mismatch

The results of comparative studies between the least-squares-based fault classification algorithm and the adaptive FDI algorithm for the case of no model mismatch are shown in Table 6 to Table 18. The percentages of correct diagnosis of the two algorithms at various fault conditions are compared. Note that "ND" represents the case that a fault is not detectable. For instance, Table 8 shows the case of a fan component fault at various fault scenarios. Specifically, each of the fan efficiency loss and flow loss health parameters is varied from 1% to 7%, and the isolation performances of the two algorithms are compared at each of the 49 fault scenarios. The cases of other 12 fault types are shown in the remaining tables. As we can see, in general, the nonlinear adaptive FDI method gives better performance. In particular, the adaptive FDI has better isolation results for the fan fault, LPC fault, VSV fault, and NC sensor fault. Some representative case studies are described below.

Figure 12 shows the case of a fan component fault (i.e., fault type 1) with an efficiency loss of 6% and a flow capacity loss of 3%. Specifically, the fault isolation results and fault magnitude estimates of the nonlinear adaptive method and linear least-squares-based fault classification method are shown in Figure 12(a) and Figure 12(b), respectively. As can be seen from Figure 12(a), the fault is correctly isolated as fault type 1 (with an accuracy of over 97%) by the adaptive FDI method, while the linear least-squares-based method often misclassifies the fault as a HPC fault (fault type 3), as shown in Figure 12(b).

(a) Nonlinear Adaptive FDI method



(b) Linear-squares-based fault classification method

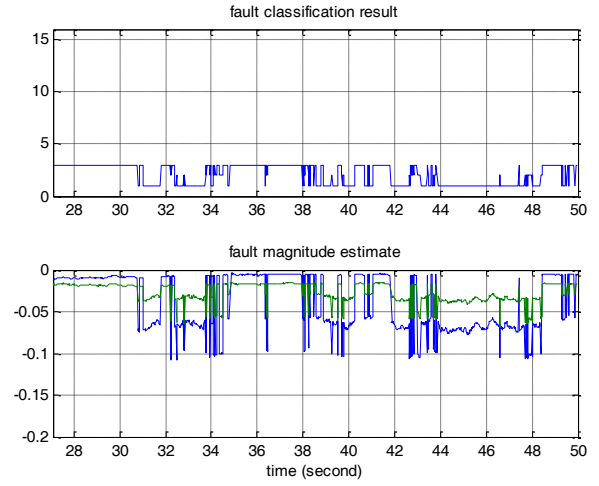
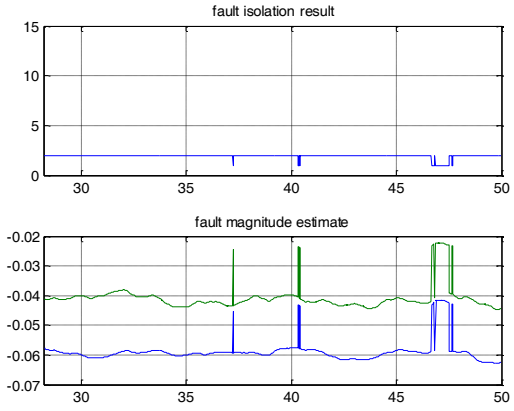


Figure 12: Case of a fan component fault with an efficiency loss of 6% and a flow capacity loss of 3%

Figure 13 shows the case of a LPC component fault (i.e., fault type 2) with an efficiency loss of 6% and a flow capacity loss of 4%. Specifically, the fault isolation results and fault magnitude estimates of the nonlinear adaptive method and linear least-squares-based fault classification method are given in Figure 13(a) and Figure 13(b), respectively. As can be seen from Figure 13(a), the fault is correctly isolated as fault type 2 (with an accuracy of about 97%) by the adaptive FDI method, while the linear least-squares-based method is oftern confused with other types of faults (see Figure 13(b)).

(a) Nonlinear Adaptive FDI method



(b) Linear-squares-based fault classification method

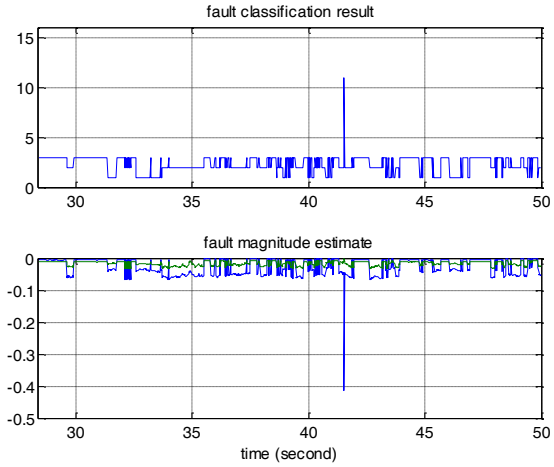
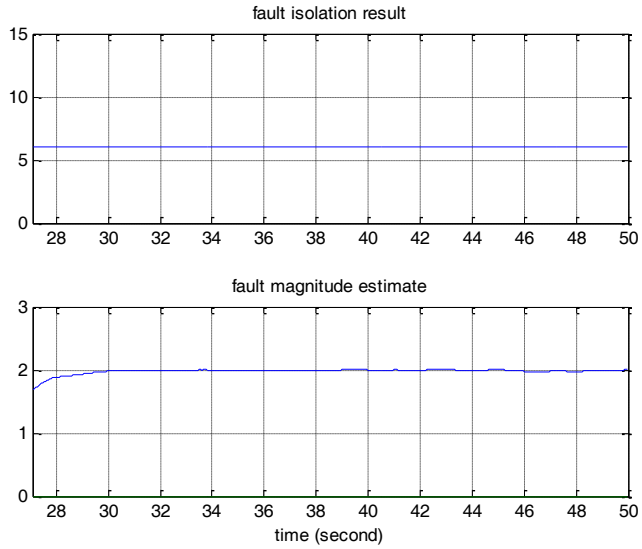


Figure 13: Case of a LPC component fault with an efficiency loss of 6% and a flow capacity loss of 4%

Figure 14 shows the case of a VSV component fault (i.e., fault type 6) with a bias of 2. Specifically, the fault isolation results and fault magnitude estimates of the nonlinear adaptive method and linear least-squares-based fault classification method are given in Figure 14(a) and Figure 14(b), respectively. As can be seen from Figure 14(a), the fault is correctly isolated as fault type 6 by the adaptive FDI method, while the linear least-squares-based method has much more misclassifications (see Figure 14(b)).

(a) Nonlinear Adaptive FDI method



(b) Linear-squares-based fault classification method

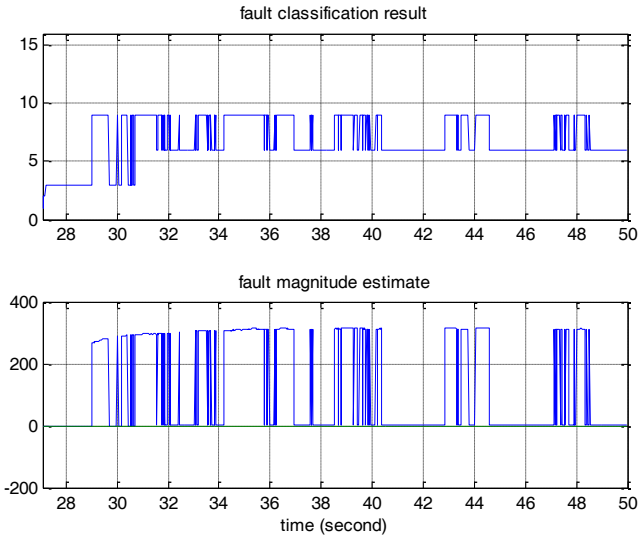
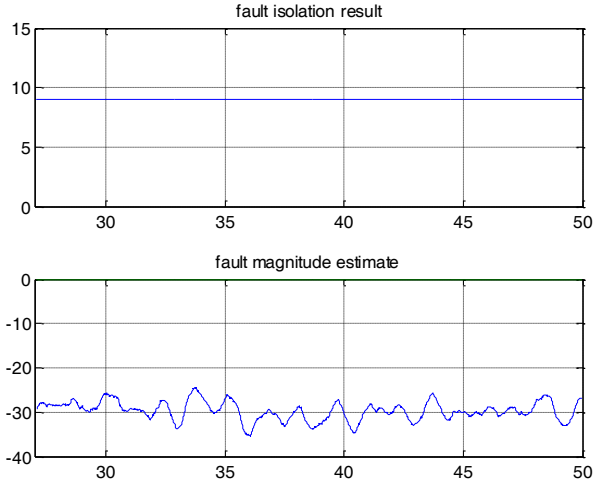


Figure 14: Case of a VSV actuator fault with a bias of 2

Figure 15 shows the case of a Nc component fault (i.e., fault type 9) with a bias of -30. Specifically, the fault isolation results and fault magnitude estimates of the nonlinear adaptive method and linear least-squares-based fault classification method are given in Figure 15(a) and Figure 15(b), respectively. As can be seen from Figure 15(a), the fault is correctly isolated as fault type 9 by the adaptive FDI method, while the linear least-squares-based method is confused with other types of faults (see Figure 15(b)).

(a) Nonlinear Adaptive FDI method



(b) Linear-squares-based fault classification method

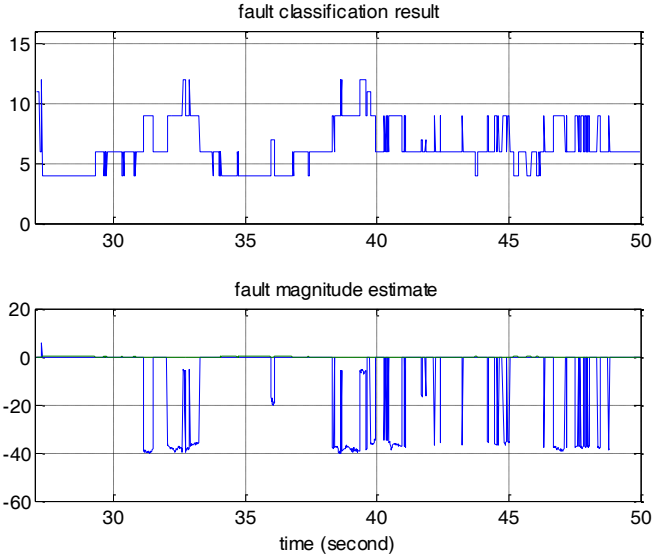


Figure 15: Case of a Nc sensor fault with a bias of -30

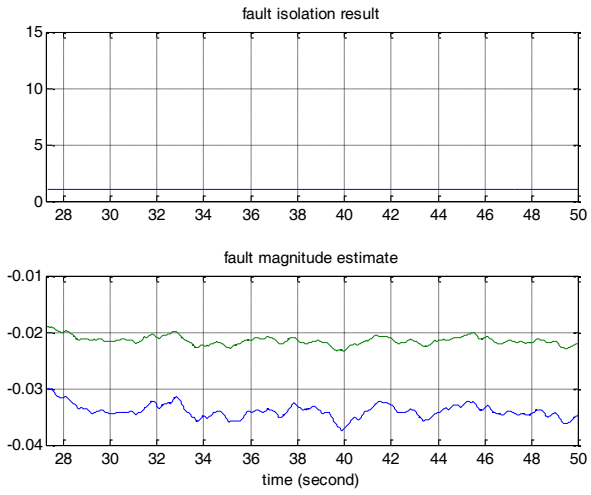
5.2.2 Case With Modeling Mismatch

The results of comparative studies for the case of 0.2% deviation in each health parameter are shown in

Table 21 to Table 33. As we can see, in general, the nonlinear adaptive FDI method gives better performance. In particular, the adaptive FDI has better isolation results for the fan fault, LPC fault, VSV fault, Nf sensor fault, and NC sensor fault. Some representative case studies are described below.

Figure 16 shows the case of a fan component fault (i.e., fault type 1) with an efficiency loss of 3% and a flow capacity loss of 2%. Specifically, the fault isolation results and fault magnitude provided by these two FDI methods are shown in Figure 16(a) and Figure 16(b), respectively. As can be seen from Figure 16(a), the fault is correctly isolated as fault type 1 by the nonlinear adaptive FDI method, while the linear least-squares-based method often misclassifies the fault as a HPC fault (i.e., fault type 3) or Ps30 sensor fault (i.e., fault type 12), as shown in Figure 16(b).

(a) Nonlinear Adaptive FDI method



(b) Linear-squares-based fault classification method

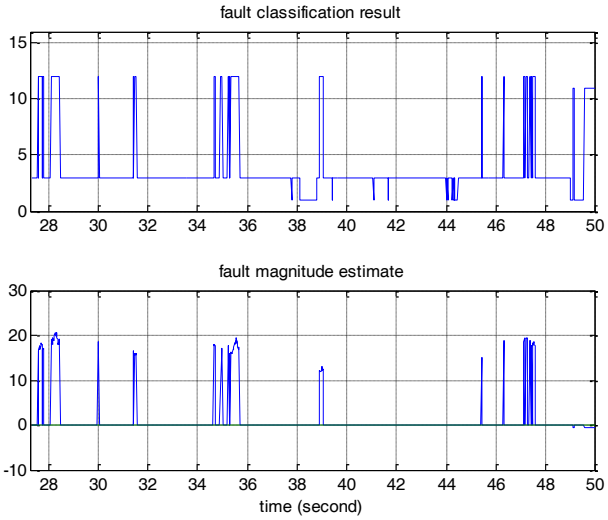
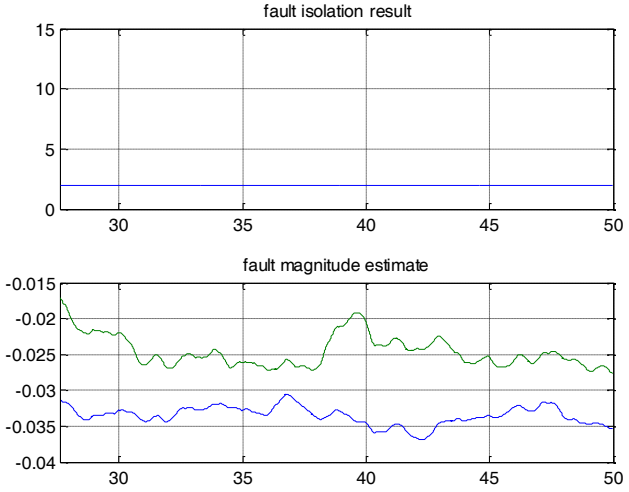


Figure 16: Case of a fan component fault with an efficiency loss of 3% and a flow capacity loss of 2%

Figure 17 shows the case of a LPC component fault (i.e., fault type 2) with an efficiency loss of 1% and a flow capacity loss of 4%. Specifically, the fault isolation results and

fault magnitude estimates of the nonlinear adaptive method and linear least-squares-based fault classification method are given in Figure 13(a) and Figure 17(b), respectively. From Figure 17(a), we can see that the fault is correctly isolated as fault type 2 by the adaptive FDI method, while the linear least-squares-based method is confused with other types of faults incluidng Ps30 sensor fault (fault type 12) and HPT fault (fault type 4), as shown in Figure 17(b).

(a) Nonlinear Adaptive FDI method



(b) Linear-squares-based fault classification method

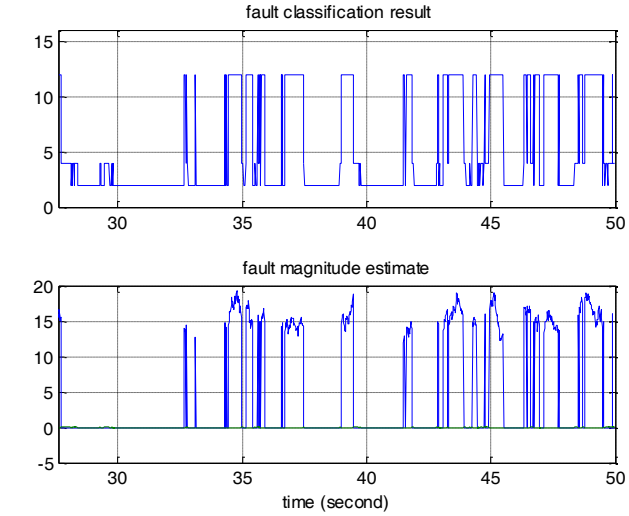
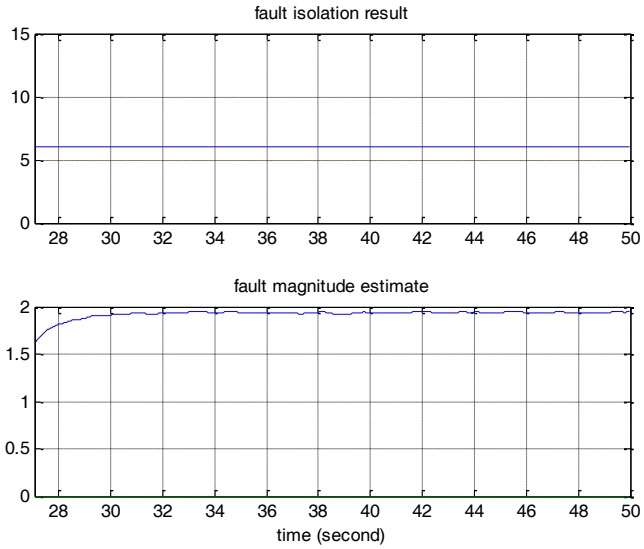


Figure 17: Case of a LPC component fault with an efficiency loss of 1% and a flow capacity loss of 4%

Figure 18 shows the case of a VSV component fault (i.e., fault type 6) with a bias of 2. As can be seen from Figure 18 (a), the fault is correctly isolated as fault type 6 by the adaptive FDI method, while the linear least-squares-based method always classifies the fault as other faults (see Figure 18(b)).

(a) Nonlinear Adaptive FDI method



(b) Linear-squares-based fault classification method

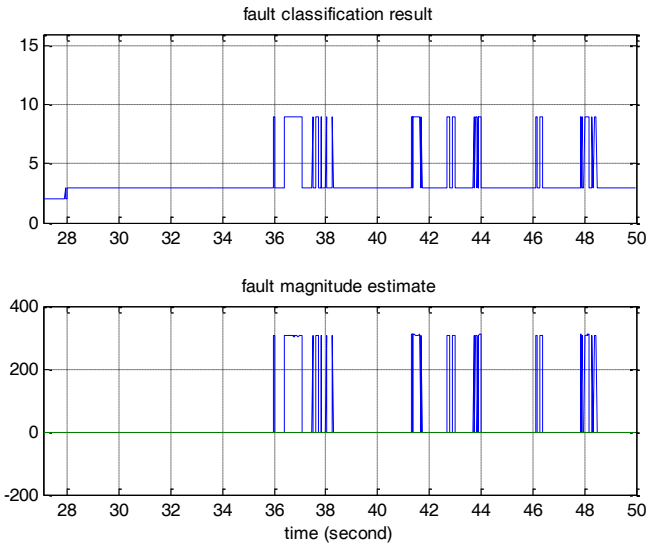
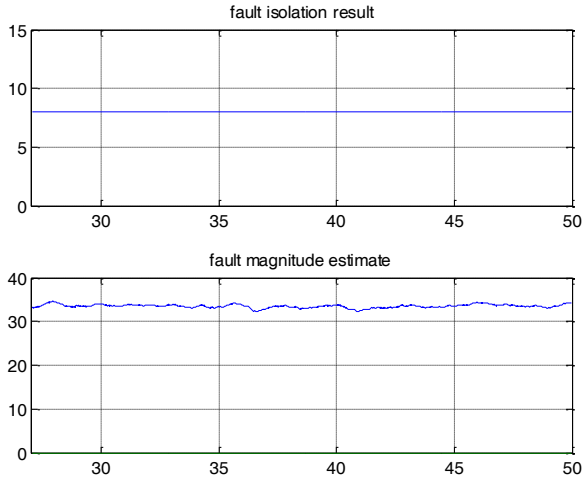


Figure 18: Case of a VSV actuator fault with a bias of 2

Figure 19 shows the case of a Nf sensor fault (i.e., fault type 8) with a bias of 27. As can be seen from Figure 19(a), the fault is always correctly isolated by the adaptive FDI method, while the linear least-squares-based method approximately has a misclassificaiton rate of 47% (see Figure 19 (b)).

(a) Nonlinear Adaptive FDI method



(b) Linear-squares-based fault classification method

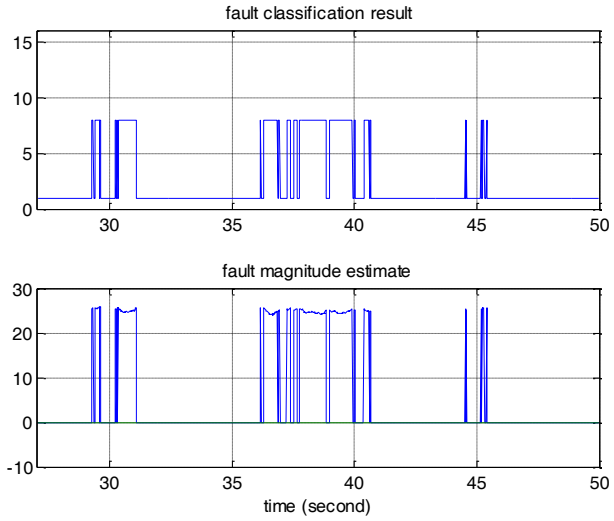
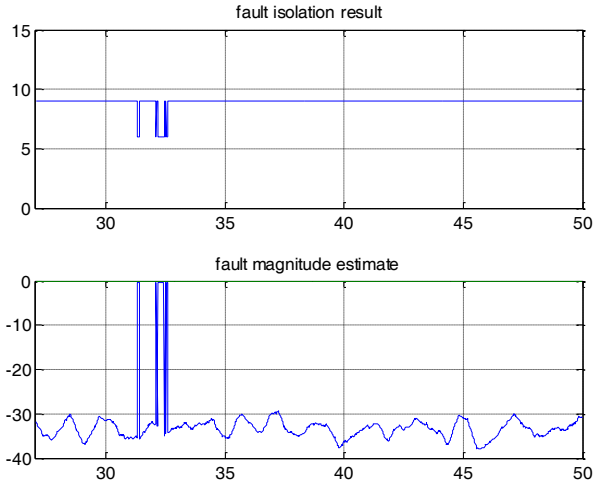


Figure 19: Case of a Nf sensor fault with a bias of 27

Figure 20 shows the case of a Nc senosr fault (i.e., fault type 9) with a bias of -50. As can be seen from Figure 20 (a), the fault is correctly isolated as fault type 9 (with an accuracy of over 97%) by the nonlinear adaptive FDI method, while the linear least-squares-based method is always confused with other types of faults (see Figure 20 (b)).

(a) Nonlinear Adaptive FDI method



(b) Linear-squares-based fault classification method

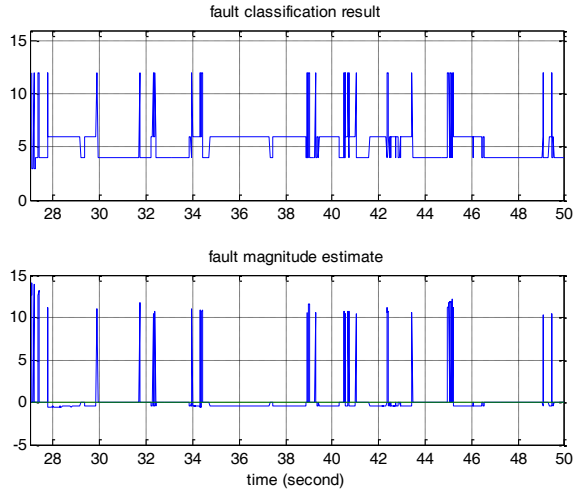


Figure 20: Case of a Nc sensor fault with a bias of -50

Conclusion

In this report, a nonlinear adaptive estimation based FDI method is designed for detecting and isolating engine sensor faults, actuator faults, and components faults. The FDI architecture has been integrated with the NASA's Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) engine model. Extensive simulation results have shown the effectiveness of the proposed FDI method.

From the observed results a few potential topics for future research stand out. First and foremost, it would not only be interesting but also beneficial to validate the method, using real engine data. Furthermore, of special attention should be paid to the magnitude of the model uncertainty bound as described in (14). Specifically, as the normal engine degradation changes, the $\bar{\eta}$ term grows, which decreases the fault sensibility of the diagnostic method. A promising approach is the use of an adaptive engine model (*i.e.* neural network), which could periodically learn the new engine dynamics, due to normal engine health degradation.

7. Appendix

7.1 Case without Modeling Uncertainty

Table 8**:** Case of Fan component fault

(a) Least-squares-based fault classification method

Eff./Flow	1%	2%	3%	4%	5%	6%	7%
1%	ND	87	86	96	95	96	100
2%	ND	99	97	100	100	99	100
3%	98	88	100	100	100	100	100
4%	100	44	100	100	100	100	100
5%	82	100	100	100	100	100	100
6%	100	100	47	100	100	100	100
7%	100	100	100	98	100	100	100

(b) Adaptive FDI method

Eff./Flow	1%	2%	3%	4%	5%	6%	7%
1%	ND	100	100	100	100	100	100
2%	ND	100	100	100	100	100	100
3%	100	100	100	100	100	100	100
4%	100	88	100	100	100	100	100
5%	100	100	100	100	100	100	100
6%	100	100	92	100	100	100	100
7%	100	100	100	100	100	100	100

Table 9**:** Case of LPC component fault

(a) Least-squares-based fault classification method

Eff./Flow	1%	2%	3%	4%	5%	6%	7%
1%	ND	ND	ND	77	98	97	99
2%	ND	ND	ND	93	97	99	100
3%	ND	ND	ND	ND	97	100	100
4%	ND	ND	ND	95	95	97	100
5%	56	ND	13	75	97	100	100
6%	86	72	27	55	84	99	100
7%	62	87	75	24	87	95	100

(b) Adaptive FDI method

Eff./Flow	1%	2%	3%	4%	5%	6%	7%
1%	ND	ND	ND	100	100	100	100
2%	ND	ND	ND	100	100	100	100
3%	ND	ND	ND	ND	100	100	100
4%	ND	ND	ND	100	100	100	100
5%	100	ND	42	100	100	100	100
6%	100	100	75	97	100	100	100
7%	100	100	99	36	100	100	100

Table 10**:** Case of HPC component fault

(a) Least-squares-based fault classification method

Eff./Flow	1%	2%	3%	4%	5%	6%	7%
1%	100	88	58	90	90	92	94
2%	100	100	100	98	67	83	88
3%	100	100	100	100	100	100	98
4%	100	100	100	100	100	100	100
5%	100	100	100	100	100	100	100
6%	100	100	100	100	100	100	100
7%	100	100	100	100	100	100	100

(b) Adaptive FDI method

Eff./Flow	1%	2%	3%	4%	5%	6%	7%
1%	100	100	100	100	100	100	100
2%	100	100	100	100	100	100	100
3%	100	100	100	100	100	100	100
4%	100	100	100	100	100	100	100
5%	100	100	100	100	100	100	100
6%	100	100	100	100	100	100	100
7%	100	100	100	100	100	100	100

Table 11: Case of HPT component fault

(a) Least-squares-based fault classification method

Eff./Flow	1%	2%	3%	4%	5%	6%	7%
1%	100	100	100	100	100	100	100
2%	100	100	100	100	100	100	100
3%	100	100	100	100	100	100	100
4%	100	100	100	100	100	100	100
5%	100	100	100	100	100	100	100
6%	99	100	100	100	100	100	100
7%	98	100	100	100	100	100	100

(b) Adaptive FDI method

Eff./Flow	1%	2%	3%	4%	5%	6%	7%
1%	100	100	100	100	100	100	100
2%	100	100	100	100	100	100	100
3%	100	100	100	100	100	100	100
4%	100	100	100	100	100	100	100
5%	100	100	100	100	100	100	100
6%	100	100	100	100	100	100	100
7%	100	100	100	100	100	100	100

Table 12**:** Case of LPT component fault

(a) Least-squares-based fault classification method

Eff./Flow	1%	2%	3%	4%	5%	6%	7%
1%	100	100	100	100	100	100	100
2%	100	100	100	100	100	100	100
3%	100	100	100	100	100	100	100
4%	100	100	100	100	100	100	100
5%	100	100	100	100	100	100	100
6%	100	100	100	100	100	100	100
7%	99	100	100	100	100	100	100

(b) Adaptive FDI method

Eff./Flow	1%	2%	3%	4%	5%	6%	7%
1%	100	100	100	100	100	100	100
2%	100	100	100	100	100	100	100
3%	100	100	100	100	100	100	100
4%	100	100	100	100	100	100	100
5%	100	100	100	100	100	100	100
6%	100	100	100	100	100	100	100
7%	100	100	100	100	100	100	100

Table 13**:** Case of VSV actuator fault

(a) Least-squares-based fault classification method

Fault Mag.	-3	-2	-1	1	2	3
Isolation Perc.	89	85	72	53	55	48

(b) Adaptive FDI method

Fault Mag.	-3	-2	-1	1	2	3
Isolation Perc.	100	100	100	100	100	100

Table 14**:** Case of Wf actuator fault

(a) Least-squares-based fault classification method

Fault Mag.	-200	-160	-120	-80	-40	40	80	120	160	200
Isolation Perc.	100	100	100	100	100	100	100	100	100	100

(b) Adaptive FDI method

Fault Mag.	-200	-160	-120	-80	-40	40	80	120	160	200
Isolation Perc.	100	100	100	100	100	97	100	100	100	100

Table 15**:** Case of Nf sensor fault

(a) Least-squares-based fault classification method

Fault Mag.	-90	-81	-72	-63	-54	-45	-36	-27	-18	-9
Isolation Perc.	100	100	100	100	100	100	100	100	100	91

Fault Mag.	9	18	27	36	45	54	63	72	81	90
Isolation Perc.	69	92	99	100	100	100	100	100	100	100

(b) Adaptive FDI method

Fault Mag.	-90	-81	-72	-63	-54	-45	-36	-27	-18	-9
Isolation Perc.	100	100	100	100	100	100	100	100	100	100

Fault Mag.	9	18	27	36	45	54	63	72	81	90
Isolation Perc.	100	100	100	100	100	100	100	100	100	100

Table 16**:** Case of Nc sensor fault

(a) Least-squares-based fault classification method

Fault Mag.	-100	-90	-80	-70	-60	-50	-40	-30	-20	-10
Isolation Perc.	2	5	1	15	12	16	40	21	ND	ND

Fault Mag.	10	20	30	40	50	60	70	80	90	100
Isolation Perc.	ND	ND	33	40	39	49	44	75	63	46

(b) Adaptive FDI method

Fault Mag.	-100	-90	-80	-70	-60	-50	-40	-30	-20	-10
Isolation Perc.	100	100	100	100	100	100	100	100	ND	ND

Fault Mag.	10	20	30	40	50	60	70	80	90	100
Isolation Perc.	ND	ND	89	75	84	97	100	100	100	98

Table 17**:** Case of T24 sensor fault

(a) Least-squares-based fault classification method

Fault Mag.	-20	-18	-16	-14	-12	-10	-8	-6	-4	-2
isolation Perc.	100	100	100	100	100	100	100	100	100	85

Fault Mag.	2	4	6	8	10	12	14	16	18	20
isolation Perc.	85	100	100	100	100	100	100	100	100	100

(b) Adaptive FDI method

Fault Mag.	-20	-18	-16	-14	-12	-10	-8	-6	-4	-2
Isolation Perc.	100	100	100	100	100	100	100	100	100	100

Fault Mag.	2	4	6	8	10	12	14	16	18	20
Isolation Perc.	100	100	100	100	100	100	100	100	100	100

Table 18**:** Case of T48 sensor fault

(a) Least-squares-based fault classification method

Fault Mag.	-56	-49	-42	-35	-28	-21	-14	-7
Isolation Perc.	100	100	100	100	100	100	100	92

Fault Mag.	7	14	21	28	35	42	49	56
Isolation Perc.	77	98	100	100	100	100	100	100

(b) Adaptive FDI method

Fault Mag.	-56	-49	-42	-35	-28	-21	-14	-7
Isolation Perc.	100	100	100	100	100	100	100	100

Fault Mag.	7	14	21	28	35	42	49	56
Isolation Perc.	100	100	100	100	100	100	100	100

Table 19**:** Case of Ps30 sensor fault

(a) Least-squares-based fault classification method

Fault Mag.	-9	-7.5	-6	-4.5	-3	-1.5	1.5	3	4.5	6	7.5	9
Isolation Perc.	100	100	100	100	100	100	100	100	100	100	100	100

(b) Adaptive FDI method

Fault Mag.	-9	-7.5	-6	-4.5	-3	-2	2	3	5	6	7.5	9
Isolation Perc.	100	100	100	100	100	100	95	100	100	100	100	100

Table 20**:** Case of T30 sensor fault

(a) Least-squares-based fault classification method

Fault Mag.	-30	-27	-24	-21	-18	-15	-12	-9	-6	-3
Isolation Perc.	100	100	100	100	100	100	100	99	95	78

Fault Mag.	3	6	9	12	15	18	21	24	27	30
Isolation Perc.	59	93	98	100	100	100	100	100	100	100

(b) Adaptive FDI method

Fault Mag.	-30	-27	-24	-21	-18	-15	-12	-9	-6	-3
Isolation Perc.	100	100	100	100	100	100	100	100	100	100

Fault Mag.	3	6	9	12	15	18	21	24	27	30
Isolation Perc.	100	100	100	100	100	100	100	100	100	100

7.2 Case with Modeling Uncertainty

Table 21**:** Case of Fan component fault

(a) Least-squares-based fault classification method

Eff./Flow	1%	2%	3%	4%	5%	6%	7%
1%	0	100	100	100	100	100	100
2%	0	85	100	100	100	100	100
3%	24	7	98	100	100	100	100
4%	97	0	95	100	100	100	100
5%	100	39	17	100	100	100	100
6%	100	87	0	96	100	100	100
7%	100	99	19	11	100	100	100

(b) Adaptive FDI method

Eff./Flow	1%	2%	3%	4%	5%	6%	7%
1%	100	100	100	100	100	100	100
2%	100	100	100	100	100	100	100
3%	100	100	100	100	100	100	100
4%	100	100	100	100	100	100	100
5%	100	100	98	100	100	100	100
6%	100	100	93	100	100	100	100
7%	100	100	100	100	100	100	100

Table 22: Case of LPC component fault

(a) Least-squares-based fault classification method

Eff./Flow	1%	2%	3%	4%	5%	6%	7%
1%	ND	2	39	58	70	94	97
2%	0	0	11	68	77	92	99
3%	0	0	0	44	86	90	97
4%	1	0	0	9	62	85	98
5%	2	0	0	1	61	90	99
6%	0	0	0	0	14	83	100
7%	13	0	0	0	2	58	94

(b) Adaptive FDI method

Eff./Flow	1%	2%	3%	4%	5%	6%	7%
1%	ND	0	63	92	100	100	100
2%	4	0	0	61	100	100	100
3%	27	0	0	44	100	100	100
4%	100	0	0	0	31	100	100
5%	100	0	0	0	0	98	100
6%	100	3	0	0	0	47	100
7%	100	21	0	0	0	0	100

Table 23: Case of HPC component fault

(a) Least-squares-based fault classification method

Eff./Flow	1%	2%	3%	4%	5%	6%	7%
1%	100	100	98	41	84	88	90
2%	100	100	100	100	100	43	52
3%	100	100	100	100	100	100	100
4%	100	100	100	100	100	100	100
5%	100	100	100	100	100	100	100
6%	100	100	100	100	100	100	100
7%	100	100	100	100	100	100	100

(b) Adaptive FDI method

Eff./Flow	1%	2%	3%	4%	5%	6%	7%
1%	100	100	100	100	100	100	100
2%	100	100	100	100	100	100	100
3%	100	100	100	100	100	100	100
4%	100	100	100	100	100	100	100
5%	100	100	100	100	100	100	100
6%	100	100	100	100	100	100	100
7%	100	100	100	100	100	100	100

Table 24: Case of HPT component fault

(a) Least-squares-based fault classification method

Eff./Flow	1%	2%	3%	4%	5%	6%	7%
1%	100	100	100	100	100	100	100
2%	100	100	100	100	100	100	100
3%	100	100	100	100	100	100	100
4%	100	100	100	100	100	100	100
5%	100	100	100	100	100	100	100
6%	100	100	100	100	100	100	100
7%	100	100	100	100	100	100	100

(b) Adaptive FDI method

Eff./Flow	1%	2%	3%	4%	5%	6%	7%
1%	99	100	100	100	100	100	100
2%	100	100	100	100	100	100	100
3%	100	100	100	100	100	100	100
4%	100	100	100	100	100	100	100
5%	100	100	100	100	100	100	100
6%	100	100	100	100	100	100	100
7%	100	100	100	100	100	100	100

Table 25: Case of LPT component fault

(a) Least-squares-based fault classification method

Eff./Flow	1%	2%	3%	4%	5%	6%	7%
1%	74	100	100	100	100	100	100
2%	80	100	100	100	100	100	100
3%	86	100	100	100	100	100	100
4%	69	98	100	100	100	100	100
5%	87	97	100	100	100	100	100
6%	73	98	100	100	100	100	100
7%	74	97	100	100	100	100	100

(b) Adaptive FDI method

Eff./Flow	1%	2%	3%	4%	5%	6%	7%
1%	85	100	100	100	100	100	100
2%	77	100	100	100	100	100	100
3%	81	100	100	100	100	100	100
4%	95	100	100	100	100	100	100
5%	100	100	100	100	100	100	100
6%	99	100	100	100	100	100	100
7%	100	100	100	100	100	100	100

Table 26**:** Case of VSV actuator fault

(a) Least-squares-based fault classification method

Fault Mag.	-3	-2	-1	1	2	3
Isolation Perc.	93	96	92	0	0	0

(b) Adaptive FDI method

Fault Mag.	-3	-2	-1	1	2	3
Isolation Perc.	100	100	67	71	100	100

Table 27**:** Case of Wf actuator fault

(a) Least-squares-based fault classification method

Fault Mag.	-200	-160	-120	-80	-40	40	80	120	160	200
Isolation Perc.	100	100	100	100	65	34	98	100	100	100

(b) Adaptive FDI method

Fault Mag.	-200	-160	-120	-80	-40	40	80	120	160	200
Isolation Perc.	100	100	100	100	100	53	100	100	100	100

Table 28**:** Case of Nf sensor fault

(a) Least-squares-based fault classification method

Fault Mag.	-90	-81	-72	-63	-54	-45	-36	-27	-18	-9
Isolation Perc.	100	100	100	100	100	100	100	99	77	6

Fault Mag.	9	18	27	36	45	54	63	72	81	90
Isolation Perc.	0	0	36	49	53	70	91	95	97	97

(b) Adaptive FDI method

Fault Mag.	-90	-81	-72	-63	-54	-45	-36	-27	-18	-9
Isolation Perc.	100	100	100	100	100	100	100	100	100	100

Fault Mag.	9	18	27	36	45	54	63	72	81	90
Isolation Perc.	20	90	100	100	100	100	100	100	100	100

Table 29**:** Case of Nc sensor fault

(a) Least-squares-based fault classification method

Fault Mag.	-100	-90	-80	-70	-60	-50	-40	-30	-20	-10
Isolation Perc.	0	0	0	0	0	0	0	0	0	ND

Fault Mag.	10	20	30	40	50	60	70	80	90	100
Isolation Perc.	ND	ND	1	0	0	0	1	0	1	2

(b) Adaptive FDI method

Fault Mag.	-100	-90	-80	-70	-60	-50	-40	-30	-20	-10
Isolation Perc.	100	100	100	100	98	97	87	88	43	ND

Fault Mag.	10	20	30	40	50	60	70	80	90	100
Isolation Perc.	ND	ND	55	54	48	64	67	70	91	89

Table 30**:** Case of T24 sensor fault

(a) Least-squares-based fault classification method

Fault Mag.	-20	-18	-16	-14	-12	-10	-8	-6	-4	-2
isolation Perc.	100	100	100	100	100	100	92	50	13	2

Fault Mag.	2	4	6	8	10	12	14	16	18	20
isolation Perc.	2	79	99	100	100	100	100	100	100	100

(b) Adaptive FDI method

Fault Mag.	-20	-18	-16	-14	-12	-10	-8	-6	-4	-2
Isolation Perc.	100	100	100	100	100	100	100	100	100	100

Fault Mag.	2	4	6	8	10	12	14	16	18	20
Isolation Perc.	100	100	100	100	100	100	100	100	100	100

Table 31**:** Case of T48 actuator fault

(c) Least-squares-based fault classification method

Fault Mag.	-56	-49	-42	-35	-28	-21	-14	-7
Isolation Perc.	100	100	100	100	100	92	18	ND

Fault Mag.	7	14	21	28	35	42	49	56
Isolation Perc.	8	55	87	100	100	100	100	100

(d) Adaptive FDI method

Fault Mag.	-56	-49	-42	-35	-28	-21	-14	-7
Isolation Perc.	100	100	100	100	100	100	81	ND

Fault Mag.	7	14	21	28	35	42	49	56
Isolation Perc.	71	100	100	100	100	100	100	100

Table 32**:** Case of Ps30 sensor fault

(a) Least-squares-based fault classification method

Fault Mag.	-9	-7.5	-6	-4.5	-3	-1.5	1.5	3	4.5	6	7.5	9
Isolation Perc.	100	100	100	100	100	98	ND	100	100	100	100	100

(b) Adaptive FDI method

Fault Mag.	-9	-7.5	-6	-4.5	-3	-2	2	3	5	6	7.5	9
Isolation Perc.	100	100	100	100	100	91	ND	100	100	100	100	100

Table 33**:** Case of T30 sensor fault

(a) Least-squares-based fault classification method

Fault Mag.	-30	-27	-24	-21	-18	-15	-12	-9	-6	-3
Isolation Perc.	100	100	100	100	100	85	76	30	9	1

Fault Mag.	3	6	9	12	15	18	21	24	27	30
Isolation Perc.	0	1	19	61	96	93	100	100	100	100

(b) Adaptive FDI method

Fault Mag.	-30	-27	-24	-21	-18	-15	-12	-9	-6	-3
Isolation Perc.	100	100	100	100	100	100	100	100	100	27

Fault Mag.	3	6	9	12	15	18	21	24	27	30
Isolation Perc.	94	100	100	100	100	100	100	100	100	100

Works Cited

1. DeCastro, J A, D K Frederick, and J S Litt. "A modular aero-propulsion system simulation of a large commercial aircraft engine." *The 44th AIAA/ASME/SAE.ASEE Joint Propuslion Conference & Exhbit* (AIAA), 2008.
2. Isermann, R. *"Fault-Diagnosis Systems" An Introduction From Fault Detection to Fault Tolerance.* Heidelberg, Berlin: Springer, 2006.
3. X. Zhang, L. Tang, "Robust Fault Diagnosis of Aircraft Engines: A Nonlinear Adaptive Estimation Based Approach", *IEEE Transactions on Control Systems Technology*, Submitted May-2011.
4. X. Zhang, M. M. Polycarpou, T. Parisini, "A robust detection and isolation scheme for abrupt and incipient faults in nonlinear systems", *IEEE Trans. on Automatic Control*, vol. 47, 2002
5. T. Kobayashi, D. Simon, "Integration of on-line and off-line diagnostic algorithms for aircraft engine health management", *2007 ASME Turbo Expo*, Montreal Canada, May 2007
6. G.P. Sallee, "Performance deterioration based on existing (historical) data", *JT9D Jet Engine Diagnostics Program*, NASA Lewis Research Center, 20??
7. Merrill, Walter C, John C Delaat, and Mahmood Abdelwahab. "Turbofan engine demonstration of sensor failure detection." *AIAA Journal of Guidance Control and Dynamics* 14.2 (1991) : 337-349. Print.
8. R. J. Patton, J. Che, "Detection of faulty sensors in Aero Jet engine systems using robust model-based methods", University of York, YORK, YO1 5DD

9. Ding, Steven X. “Model-Based Fault Diagnosis Techniques Design Schemes, Algorithms, and Tools”. Berlin: Springer, 2008. <*http://public.eblib.com/EBLPublic/PublicView.do?ptiID=372301* >

10. Isermann, R. (2011). *Fault-Diagnosis Applications: Model-Based Condition Monitoring:Actuators, drives, machinery,plants, sensors, and fault-tolerant systems.* Springer.

11. Isermann, R. “Fault Diagnosis Systems” *An Introduction from Fault Detection to Fault Tolerance*, Berlin-Springer, 2006.

12. C. Edwards T. Lombaerts and H.Smailli, “Fault Tolerant Fight Control: A Benchmark Challenge” *Lecture Notes in Control and Information Sciences*, Vol. 339 Springer, 2010. Print.

13. D.Simon, F Bird, C. Davidson, A. Volponi and R.E. Iverson “Benchmarking Gas Path Diagnostics Methods: A Public Approach”, *Turbo Expo 2008 - Gas Turbine Technical Congress and Exposition*, Berlin, Germany June9-13, 2008.

14. Merrill, W.C., DeLaat, J.C. and Brutton, W. M., “Advanced detection isolation and accomodation of sensor failures - a real-time evaluation”, J. Guidance Control & Dynamics, 11 (6), 1988.

15. Merrill W. C.,”Sensor Failure detection for jet engines”, Control & Dynamic Systems - Advances in Aerospace Systems and Control Systems, Vol. 33, Academic Press, Inc, 1990.

16. Beattie, E. C., La Prad, R. F., McGlone, M.E., Rock, S.M. and Ahkter, M.M., “Sensor failure detection for jet engines”, NASA-CR-168190, May 1981.

17. Merrill, W.C., DeLaat, J.C.m Kroszenszkewicz, S.N. and Abdelwahab, M. "Full-scale engine demonstration of an advanced sensor failure detection, isolation and accommodation algorithm - preliminary results:, AIAA Guidance, Navigation and Control Conference, Monterey, California, 17-19 August 1987

18. Merrill, W. C., Guo, T.H. DeLaat, J.C. and Duyar, A. "Real-time fault diagnosis for propulsion system", Preprints of IFAC/IMACS Symposium SAFEPROCESS'91, Baden-Baden, Sept 10-13, 1991

19. T. Kobayashi, D.L. Simon and J.S. Litt "Application of a Constant Gain Extended Kalman Filter for In-Flight Estimation of Aircraft Engine Performance Parameters", NASA Scientific and Technical Information Program, NASA/TM - 2005

20. T. Kobayashi, D.L. Simon, "Hybrid Kalman Filter Approach for Aircraft Engine In-Flight Diagnostics: Sensor Fault Detection Case"

21. Sugiyama, N., 2000, "System Identification of Jet Engines", Journal of Engineering for Gas Turbines and Power, 122, pp 19-26.

22. T. Kobayashi, D.L.Simon, "Application of a bank of Kalman Filters for aircraft engine fault diagnostics" NASA Scientific and Technical Information Program, NASA/TM - 2005 <*http://gltrs.grc.nasa.gov*>

23. T. Kobayashi, D.L.Simon, "Evaluation of an enhanced bank of Kalman filters for In-Flight aircraft engine sensor fault diagnostics" *Turbo Expo 2004*, Vienna Austria, June 14-17 - 2004.

24. A.J. Volponi, H. DePold, R. Ganguli and C. Daguang, "The use of Kalman filter and neural network methodologies in gas turbine performance diagnostics: A

comparative study" J. Eng. Gas Turbine Power 125, 917 (2003), DOI:10.1115/1.1419016

25. M. Kurosaki, T. Morioka, K. Ebina, M. Maruyama, T. Yasuda and M. Endoh, J. Eng. Gas Turbines Power 126, 726 (2004), DOI:10.1115/1.1787515

26. D. Yixin, and K. Passino, "Stable Fault Diagnosis for a Turbine Engine", 2001. , http://www.elsevier.com/locate/conengprac

27. R.J. Patton, J. Chen, "Robust Model-based Fault Diagnosis for Dynamic Systems", Kluwer Academic Publishers, 1999.

28. NASA Vehicle Integrated Health Management (IVHM) Technical Plan, http:www.aeronautics.nasa.gov/nra_pdf/ivhm_tech_plan_c1.pdf, accessed November, 2009.

29. Blanke, M., Kinnaert, M., Lunze, J., Staroswiecki, M., and Schröder, J., 2006. *Diagnosis and Fault-Tolerant Control,* Berlin; New York : Springer, 2006.

30. Duyar, A., Eldem, V., Merrill, W., and Guo, T-H., 1994, "Fault detection and diagnosis in propulsion systems: a fault parameter estimation approach," *Journal of Guidance, Control, and Dynamics*, 17(1), pp. 104-108.

31. Farrell, J., and Polycarpou, M. M., 2006, *Adaptive Approximation Based Control: Unifying Neural, Fuzzy and Traditional Adaptive Approximation* Approaches, Wiley Interscience.

32. Frederick, D. K., DeCastro, J. A., Litt, Jonathan, S., 2007, *User's Guide for the Commercial Modular Aero-Propulsion System Simulation (C-MAPSS)*, NASA Technical Report NASA/TM-2007-215026.

33. Ioannou, P. A. and Sun, J., 1996, *Robust Adaptive Control*, Prentice Hall, Englewood Cliffs, NJ.

34. Merill, W.C., DeLaat, J.C., and Abdelwahab, M., 1991, "Turbofan engine demonstration of sensor failure detection," *AIAA Journal of Guidance, Control, and Dynamics*, Vol. 14, No. 2, pp. 337-349.

35. Patton, R., Chen, J., 1992, "Robust fault detection of jet engine sensor systems using eigenstructure assignment," *Journal of Guidance, Control, and Dynamics*, 15(6), pp. 1491-1497.

36. Rajamani, R., 1998, "Observer design for Lipschitz nonlinear systems", *IEEE Transactions on Automatic Control*, Vol. 43, No. 3, pp. 397-401.

37. Simon, D., Bird, J., Davison, C., Volponi, A., and Iverson, R. E., 2008, Benchmarking gas path diagnostic methods: a public approach, , NASA Technical Report NASA/TM-2008-215271.

38. Vemuri, A. T. and Polycarpou, M. M. , 1997, "Neural network based robust fault diagnosis in robotic systems," *IEEE Transactions on Neural Networks*, vol. 8, no. 6, pp. 1410-1420.

39. Wulf, R. H., 1980, Engine Diagnostics Program CF6-50 Engine Performance Deterioration, NASA Technical Report CR-159867.

40. Zhang, X., Polycarpou, M. M. and Parisini, T, 2008, "Design and analysis of a fault isolation scheme for a class of uncertain nonlinear systems," *IFAC Annual Reviews in Control*, pp. 107-121.

41. Zhang, X., Polycarpou, M. M. and Parisini, T, 2010, “Fault diagnosis of a class of nonlinear uncertain system with Lipschitz nonlinearities using adaptive estimation,” *Automatica*, vol. 46, pp. 290-299.